



UNIVERSITAT POLITÈCNICA DE CATALUNYA  
BARCELONATECH

Escola Politècnica Superior d'Enginyeria  
de Manresa



## Treball Final de Grau

# ***DISSENY I AUTOMATITZACIÓ D'UN CONTROL DE SISTEMA DE TANCS INTERCONNECTATS***

---

**Grau en Enginyeria Electrònica i  
Automàtica**

**Curs 17/18**

Autor: . FERRAN VICEDO LOPEZ

Data: 06/07/2018

Localitat: Manresa





Aquest treball el vull dedicar especialment als meus pares, que són els que més m'han ajudat i animat a treure'm aquest grau. Se que els hi farà molta il·lusió.

També agrair als meus germans tots els seus ànims, sempre els tinc al meu costat. Als meus amics que quan he necessitat distraurem i riurem en èpoques d'estres han estat allà.

Vull agrair totes les facilitats que m'han donat a la feina per poder acabar tranquil·lament el grau.

I per últim vull agrair a tot el professorat de la EPSEM la seva ajuda durant tots aquests anys, especialment als tutors del TFG la Teresa Escobet i al David Soler, per la seva dedicació i paciència en tot aquest temps.

Gràcies a tots!!!



## RESUM DEL PROJECTE (en català o castellà)

En el treball fi de grau realitzat s'ha dissenyat i automatitzat una estació de treball de tancs interconnectats. Per el disseny s'han utilitzat unes estacions de treball que hi havia al laboratori. Aquestes s'han desmuntat i interconnectat donant lloc a una nova estació amb la qual simular sistemes batch. La interconnexió s'ha realitzant tant a nivell de circuit d'aigua amb canonades, vàlvules de pas i dipòsits com a nivell de connexions electròniques. Per a la seva automatització s'ha utilitzat el PLC X20 CP 1381 de B&R Automation.

En el treball es descriu àmpliament l'entorn de programació d'aquest PLC, l'Automation Studio. La descripció realitzada pretén ser una guia d'usuari per poder emprar aquest programa amb certa facilitat.

Els programes implementats, per a l'automatització de l'estació dissenyada, s'han centrat en el control de nivell d'un dels dipòsits utilitzant les mesures dels sensors digitals i analògics disponibles, i actuant sobre les electrovàlvules digitals i l'analògica i la motobomba. Els programes implementats van de menys a més complexitat i inclouen la sintonia d'un regulador PID per assolir la consigna de nivell desitjada.



## RESUM DEL PROJECTE (en anglès)

This final degree project consists of designing and automating a workstation with interconnected fluid tanks. To develop this project has been used a laboratory workstation who has been rebuilt to a new configuration destined to simulate batch system. Water pipes, valves and water tank has been interconnected with the PLC X20 CP 1381 of B&R Automation.

In this work, the programming environment of this PLC, Automation Studio, is widely described. The description made aims to be a user guide to be able to use this program easily.

The implemented programs, for the automation of the designed station, have focused on the level control of one of the tanks using the measurements provided by the on/off and analog level sensors, and acting on the on/off and analog valves, and pump. These programs go from less to more complexity and include the tuning of a PID regulator to achieve the desired setpoint.



# ÍNDEX

<b>1. INTRODUCCIÓ .....</b>	<b>12</b>
1.1. SISTEMA BATCH .....	12
1.1.1. CONTROLADORS LÒGICS PROGRAMABLES .....	13
1.1.2. SISTEMES DE CONTROL DISTRIBUÏT .....	16
1.2. JUSTIFICACIÓ DEL PROJECTE .....	18
1.3. OBJECTIU .....	18
1.4. FINALITAT DEL PROJECTE .....	19
<b>2. ANTECEDENTS.....</b>	<b>21</b>
2.1. TREBALLS FI DE GRAUS ANTERIORS.....	21
2.2. LES NOSTRES ESTACIONS.....	21
<b>3. DISSENY L'ESTACIÓ DE TREBALL .....</b>	<b>25</b>
3.1. DESCRIPCIÓ GENERAL DE LA MAQUETA.....	25
3.2. ESQUEMA DE LA MAQUETA.....	25
3.3. DESCRIPCIÓ DELS SENSORS I ACTUADORS .....	27
3.3.1. Sensors de nivell.....	27
3.3.2. Sensor de pressió .....	28
3.3.3. Sensor de cabal .....	28
3.3.4. Electrovàlvules.....	28
3.3.5. Motobomba.....	29
3.4. TECLAT D'OPERACIÓ.....	29
<b>4. SISTEMA DE CONTROL I AUTOMATITZACIÓ .....</b>	<b>31</b>
4.1. X20 CP 1583 / X20 CP 1381 .....	31
4.1.1. CPU X20 CP 1381 .....	31
4.2. CONNEXIONS AMB ELS SENSORS I ACTUADORS.....	35
4.2.1. Connexió dels sensors de nivell digitals.....	35
4.2.2. Connexió de les electrovàlvules digitals.....	36



4.2.3.	Connexió motobomba i electrovàlvula V 1.5 (AN).....	37
4.2.4.	Connexió sensors analògics.....	39
4.2.5.	Connexió botonera, leds, i adquisició de dades.....	40
<b>5.</b>	<b>ENTORN DE PROGRAMACIÓ .....</b>	<b>43</b>
5.1.	INICIALITZACIÓ A L'AUTOMATION STUDIO .....	43
5.2.	PROGRAMACIÓ.....	50
5.3.	PROGRAMACIÓ DEL PLC EN LLENGUATGE LADDER .....	53
5.3.1.	Definició de variables .....	54
5.3.2.	Programació en Ladder .....	56
5.4.	SIMULACIÓ .....	61
5.5.	CONNECTATS AMB EL PLC .....	67
<b>6.</b>	<b>AUTOMATITZACIÓ DE L'ESTACIÓ .....</b>	<b>68</b>
6.1	SISTEMA D'ESDEVENIMENTS DISCRETS.....	68
6.1.1.	Seqüència bàsica .....	68
6.1.2.	Seqüència indefinida .....	69
6.1.3.	Seguretat.....	70
6.2.	AUTOMATITZACIÓ COMPLETA DE L'ESTACIÓ DE TREBALL .....	71
6.2.1.	Aplicació del PID a la nostre estació de treball.....	74
<b>7.</b>	<b>REGULACIÓ DEL NIVELL I MONITORITZACIÓ .....</b>	<b>85</b>
<b>8.</b>	<b>PROVES, PROBLEMES I SOLUCIONS.....</b>	<b>92</b>
<b>9.</b>	<b>CONCLUSIONS .....</b>	<b>93</b>
<b>10.</b>	<b>BIBLIOGRAFIA .....</b>	<b>95</b>
<b>11.</b>	<b>ANNEXOS .....</b>	<b>97</b>
11.1.	ANNEX 1 .....	97

# ÍNDEX DE FIGURES

Figura 1. Esquema de l'automatització via PLC.....	15
Figura 2. Diagrama de flux i foto del procés de control de nivell. ....	22
Figura 3. Diagrama de flux i foto del procés de control de cabal.....	22
Figura 4. Diagrama de flux i foto del procés de control de pressió.....	23
Figura 5. Esquema i foto del procés de control de temperatura.....	23
Figura 6. Esquema de l'estació 1. ....	26
Figura 7. Imatge de l'estació 1.....	27
Figura 8. Teclat d'operació. ....	29
Figura 9. X20CP1381 amb els mòduls. ....	31
Figura 10. Mòdul X1 .....	32
Figura 11. Mòdul X2.....	33
Figura 12. Mòdul X3.....	33
Figura 13 . Mòdul X4.....	34
Figura 14. Esquema elèctric de les connexions dels sensors digitals de nivell .....	35
Figura 15. Connexions al PLC.....	36
Figura 16. Esquema elèctric electrovàlvules digitals.....	36
Figura 17. Esquema de connexions de les electrovàlvules al PLC .....	37
Figura 18. Esquema elèctric de la motobomba i la V1.5.....	38
Figura 19. Connexions de la motobomba i l'electrovàlvula al PLC.....	38
Figura 20. Esquemes elèctrics sensors analògics .....	39
Figura 21. Connexions al PLC.....	40
Figura 22. Esquema elèctric de la botonera .....	42





Figura 23. Connexions adquisició de dades.....	42
Figura 24. Imatge new project.....	43
Figura 25. Imatge nom projecte .....	44
Figura 26. Identificació del PLC .....	44
Figura 27. Finestra del nostre projecte.....	45
Figura 28. Imatge de selecció de programa .....	46
Figura 29. Imatge finestres del programa.....	46
Figura 30. Physical view .....	47
Figura 31. Per forçar variable amb la sortida que volem .....	47
Figura 32. Selecció de la variable que volem mapejar amb la sortida física.....	48
Figura 33. CPU .....	49
Figura 34. Enable/disable del programa .....	50
Figura 35. ETH, configuració .....	51
Figura 36. IP adress.....	51
Figura 37. Canviar IP ordinador .....	51
Figura 38. Online -> Settings .....	52
Figura 39. Browse.....	52
Figura 40. Connect .....	53
Figura 41. OFFLINE-> RUN.....	53
Figura 42. Definició variables globals.....	54
Figura 43. Definició variables locals.....	55
Figura 44. Mapeig sensors analògics del mòdul X1 .....	55
Figura 45. Mapeig entrades digitals del mòdul X2.....	55



Figura 46. Mapeig sortides digitals del mòdul X3.....	56
Figura 47. Maig actuadors analògics del mòdul X4 .....	56
Figura 48. Contacte obert.....	57
Figura 49. SET i RESET.....	57
Figura 50. Activació bobina T1 .....	57
Figura 51. Selecció MOVE .....	58
Figura 52. MOVE.....	59
Figura 53. COMPARE .....	59
Figura 54. TON .....	60
Figura 55. Ajuda del temporitzador.....	60
Figura 56. Compilar.....	62
Figura 57. Programa compilat amb error .....	62
Figura 58. Compilat OK.....	63
Figura 59. Activar mode simulació.....	63
Figura 60. En mode RUN .....	63
Figura 61. Transfer.....	64
Figura 62. Transferir programa.....	64
Figura 63. Transferència realitzada .....	65
Figura 64. Watch.....	65
Figura 65. Add variables.....	66
Figura 66. Selecció de variables.....	66
Figura 67. Forçar variables.....	67
Figura 68. Grafset seqüència bàsica .....	69



Figura 69. Grafcet Seqüència indefinida amb aturada .....	70
Figura 70. Grafcet automàtic amb aturada i PE. ....	71
Figura 71. Diagrama de blocs d'un PID .....	72
Figura 72. Trace .....	74
Figura 73. Inster variable al trace.....	75
Figura 74. Guardar gràfica del trace .....	75
Figura 75. Valor inicial .....	76
Figura 76. Valor final.....	77
Figura 77. Càlcul del 63% de la diferencia entre el valor final i el inicial.....	78
Figura 78. MTBascisStepTunnig.....	79
Figura 79. Conversió de INTEGER a REAL.....	80
Figura 80. Conversió de REAL a INTEGER.....	80
Figura 81. Constants PID.....	81
Figura 82. Function Blocks PID.....	82
Figura 83. Grafcet PID .....	84
Figura 84. Add Visu .....	85
Figura 85. Init Page .....	85
Figura 86. Elements del Visu .....	86
Figura 87. Configuració Button .....	87
Figura 88. Configuració visualitzador numèric.....	88
Figura 89. Configuració entrada numèrica .....	88
Figura 90. Color Maps .....	89
Figura 91. Configuració del shape .....	89



Figura 92. Pantalla final.....	90
Figura 93. Configurar Visu.....	90
Figura 94. VNC viewer .....	91
Figura 95. Pantalla virtual online .....	91



## INDEX DE TAULES

Taula 1. Connexions X20CP1381 .....	32
Taula 2. Connexions mòdul X2 .....	33
Taula 3. Connexions mòdul X3 .....	34
Taula 4. Taula connexions sensors digitals.....	35
Taula 5. Connexions electrovàlvules.....	37
Taula 6. Connexions motobomba i electrovàlvula analògica .....	38
Taula 7. Connexions sensors analògics.....	40



# 1. INTRODUCCIÓ

En aquest projecte s'ha dissenyat quatre estacions que podrien simular el cas d'un procés tipus Batch, en el qual es vol controlar la distribució d'aigua en diferents dipòsits, la pressió en un tanc, el cabal que circula per un tub i el nivell d'un dipòsit.

## 1.1. Sistema Batch

El sistema de tancs interconnectats dissenyat es pot considerar un sistema del tipus batch o en lots, per aquest motiu en la introducció del treball fi de grau s'ha realitzat una cerca de com es controlen aquests tipus de sistemes.

Els sistemes Batch són aquells processos de fabricació seqüencial que es realitzen per lots. Són sistemes en els quals cal gestionar i dur a terme una fabricació acotada, en nombre d'unitats o de temps, d'un determinat producte de característiques predefinides.

Dins dels nivells de l'automatització, aquests sistemes són, en la seva majoria, sistemes de control distribuïts (DCS), incorporen sistemes de control de variables, un sistema de supervisió (SCADA) i la part de gestió i execució de la informació de la planta. En la part de control es solen utilitzar reguladors PID per a la regulació de les variables de procés (temperatures, nivells, fluxos) i PLCs pel control de maniobres. La part de supervisió és l'encarregada d'enviar les ordres cap al sistema de control. La gestió i execució de la informació de la planta s'encarrega de recollir, emmagatzemar i empaquetar la informació procedent del sistema de control durant la fabricació del Batch. Són capaços de comunicar-se amb altres sistemes de la fàbrica, com a exemple podríem citar el control de l'estoc de la matèria primera necessària per a la producció d'un lot de producte.

El sistema de control Batch és un sistema de control de processos de producció per lots i està regit per la normativa ISA.S88, que es basa en l'execució d'una recepta de producció, garantint la traçabilitat i seguretat del sistema. En un procés Batch, l'operador haurà de poder seleccionar la recepta de producció, la fórmula i poder executar de forma manual o automàtica, les diferents seqüències d'operació.

Per a realitzar el control d'aquests sistemes ens trobem amb dues tecnologies: els controladors lògics programables (PLC) i els sistemes de control distribuïts (DCS). Sempre hi ha hagut una diferència entre els PLC i els DCS, però aquestes diferències s'han retallat considerablement els últims anys. Avui en dia qualsevol planta industrial tipus batch, ja sigui gran o petita, es pot beneficiar del millor de cada una d'aquestes metodologies per a la seva automatització global.

### 1.1.1. CONTROLADORS LÒGICS PROGRAMABLES

El PLC és un dispositiu electrònic capaç d'executar lògica programable per software per a l'automatització de processos industrials, sobretot en les indústries de manufactura. La seva evolució es troba documentada en autors com (Segovia & Theorin, 2012; Clare, et al., 2006; Morley, 2018). Es van desenvolupar al entorn dels anys 60 per substituir la lògica de relés, molt utilitzada en l'automatització de sistemes lògics (on/off) de sectors com l'automotriu. El desenvolupament d'aquests dispositius va ser motivat pel fet que el canvi de producte a les línies de fabricació suposava un cost molt alt i, per tant les empreses del sector d'automoció, van reptar a les empreses de control a trobar solucions per reduir aquest cost addicional. Concretament, l'any 1968 GM Hydramatic (divisió de transmissió automàtica de General Motors) va sol·licitar una proposta per substituir electrònicament els sistemes de cablejat de relés. La proposta guanyadora va ser de Bedford Associates. El resultat del primer PLC va ser anomenat 084 per Bedford Associates. Bedford Associates va crear l'empresa Modicon (Modular Digital Controller) per desenvolupar, fabricar, vendre i realitzar el manteniment del PLC. En aquest projecte va treballar Dick Morley que és considerat el "pare" dels PLC. PLC és una marca registrada per Allen Bradley, però aquestes sigles ja son utilitzades per tothom per als PLC.

Als anys 70, es van incorporar als PLC els microprocessadors millorant les seves prestacions. Concretament aquest canvi va incidir en modernitzar la comunicació home-màquina, millorar el càlcul matemàtic i la manipulació de dades, entre altres. Els microprocessadors AMD 2901 i 2903 van ser els més emprats.

A la segona meitat de la dècada dels 70, es van millorar aspectes com: la capacitat de memòria, la incorporació de dades d'entrades/sortides remotes analògiques i numèriques, funcions de control de posició, etc. També s'afegeixen altres tipus de llenguatges al tradicional de lògica de contactes, amb la incorporació de noves instruccions i més potents. Un desenvolupament que cal destacar, és el de les comunicacions amb perifèrics, concretament amb els ordinadors. El primer sistema de comunicació va ser el bus de Modicon (Modbus).

Als anys 80 va sorgir el primer intent d'estandarditzar les comunicacions amb el protocol MAP ( Manufacturing Automation Protocol) per General Motors. També es van començar a reduir les dimensions dels PLC; es va millorar la velocitat de resposta, compactar les entrades i sortides en els seus respectius mòduls, i desenvolupar mòduls de control (PID, servo controladors).

Fins als 90, cada fabricant de PLC definia el seu propi protocol de comunicació, no essent compatibles entre ells. És a partir dels anys 90 quan comença a reduir-se l'aparició de nous protocols i comencen a definir-se els primers protocols per estandarditzar les comunicacions i la programació dels PLC. L'últim protocol d'estandardització i vigent



actualment és el IEC-1131-3, aquest unifica els protocols de programació en un estàndard internacional.

Els PLC actuals, denominats d'última generació, són més compactes, més eficients, més ràpids, amb la possibilitat d'afegir mòduls d'expansió de I/O, i tenen la possibilitat de connectar-se de manera remota. Avui en dia els PLC no només controlen la lògica de funcionament d'una màquina, plantes, i processos industrials, sinó que també poden realitzar operacions aritmètiques i, per tant, permeten treballar fàcilment amb senyals analògics per implementar estratègies de control, com els clàssics controls PID.

El primer tipus de llenguatge utilitzat en la programació de PLC va ser el *l·listat d'instruccions*, que consistia en introduir les ordres de control mitjançant una llista seqüencial de codis en llenguatge màquina. Després, per facilitar la programació, es va introduir el diagrama de contactes (LADDER) simulant el cablejat elèctric de relés per facilitar als tècnics la seva utilització.

Els PLC s'han utilitzat majoritàriament en processos de manufactura, en els quals la gran majoria de senyals d'entrada/sortida (I/O) són lògics i no presenten grans restriccions en el temps de cicle, ja que aquest pot ser de l'ordre de mil·lisegons. Amb el temps, els PLC han anat incorporant funcionalitats dels DCS, com blocs de control PID i entrades/sortides analògiques, entre altres.

Un PLC normalment està compost per una CPU, mòduls d'entrada/sortida (I/O) i interfícies de comunicació. Es caracteritza per ser escalable tant amb potència de càlcul com en quantitat de I/O. És molt versàtil, fàcil de programar i disposa de diferents llenguatges de programació marcats per la norma IEC 61131-3, millorant així la primera opció de lògica d'escala (LADDER).

La última normativa de programació de PLCs és la IEC- 61499, en aquesta s'estandarditza l'ús de blocs funcionals (FB) i permeten utilitzar el PLC com a sistema de control distribuït (Vyatkin, 2009).

També hi ha l'opció de supervisar el procés amb interfícies home-màquina (HMI), que poden ser des de panells a PLCs amb un software SCADA que permet l'adquisició de dades, supervisió i operació del procés. La connexió entre els PLCs i els HMI es realitza de forma manual (encara que actualment els programes ofereixen més suport en aquest camp) on es defineix el protocol de comunicació, s'assignen registres i les seves direccions de hardware, i es programa la comunicació.

L'estructura general d'automatització a la indústria mitjançant PLCs es mostra a la Figura 1. A la part de control es troba el PLC que és on es llegeixen els sensors, on es troben totes les comunicacions i on es dona l'ordre als actuadors, i el PLC és el que realitza tot el



control del procés. A la part operativa és on es troben tots els actuadors, preactuadors, sensors, etc... On es duen a terme totes les accions físiques que afecten el procés, com es pot veure a la Figura 1.

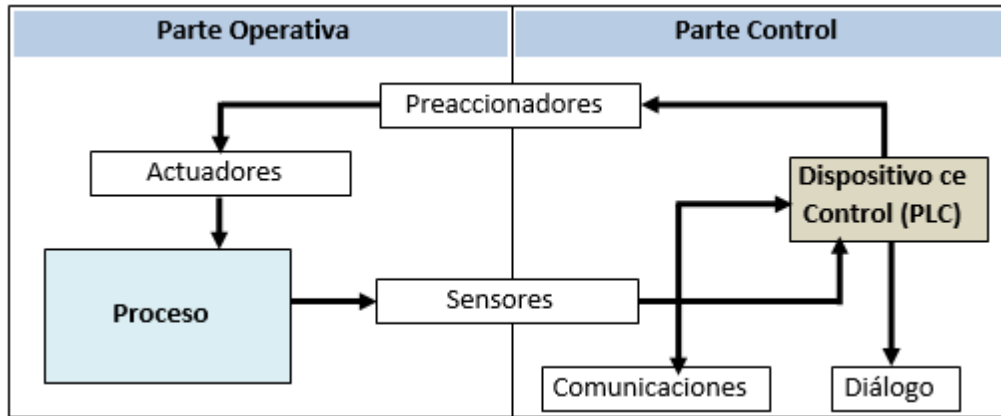


Figura 1. Esquema de l'automatització via PLC.

### Els principals avantatges de treballar amb PLC són:

- Elevat rendiment i alta fiabilitat. Ja que són capaços de controlar i automatitzar processos de gran complexitat. Són molt robustos i per tant és molt difícil que tinguin avaries.
- La més alta connectivitat per processos industrials. Molt fàcil de comunicar amb altres components de manera senzilla.
- Optimització dels temps. Tenen una alta resposta.
- Facilitat per al muntatge i la programació. El muntatge és realment senzill: simplement s'ha d'alimentar i cablejar els sensors i els actuadors. La programació està realment molt definida i amb una mica de pràctica és fàcil d'aprendre.
- Es poden realitzar modificacions sense canviar el cablejat. A l'hora de realitzar modificacions, només s'ha de modificar el programa sense haver de tocar cap connexió.
- Mínim espai d'aplicació.
- Reducció de costos.

### 1.1.2. SISTEMES DE CONTROL DISTRIBUÏT

El concepte de DCS consisteix en què tots els elements conformen un únic sistema que es controla integralment, així no hi ha la necessitat de crear interfícies entre ells, i per tant s'utilitzen eines d'enginyeria que permeten centralitzar tot el sistema des d'una única estació.

A finals de la dècada dels 60 la indústria va buscar la solució més eficient per substituir els sistemes de control basats en circuits elèctrics amb relés i interruptors, així van néixer els PLCs, però les fàbriques i els processos industrials es van tornar més complexos. La introducció de microprocessadors en el món del control de processos va propiciar que diversos fabricants desenvolupessin solucions basades en un sistema de control continu per blocs. Aquests sistemes tenien com a característica principal l'ús de diversos processadors, cadascun dels quals controlava una unitat del procés d'una planta, així que quan fallava un d'ells només es quedava sense controlar el procés d'aquesta part. És així com l'any 1974 Honeywell va ser la primera companyia en llençar al mercat el primer Sistema de Control Distribuït, el TDC 2000 (Total Distributed Control), (de 3000-20000 punts I/O) el qual es va convertir en el sistema més utilitzat a la indústria.

Originalment els DCS s'utilitzaven en plantes a gran escala, fet que comportava grans costos d'instal·lació i una enginyeria complexa. Un gran canvi dels últims 20 anys va ser substituir els hardwares propis per PCs, i les tecnologies d'àrea local (LAN) per estàndards de comunicació (Ethernet, TCP/IP), fet que va comportar una gran disminució dels costos i gran simplificació de les instal·lacions.

Actualment els DCS es poden escalar depenent de la necessitat de la instal·lació reduint el cost, i es disposa d'una norma d'estandardització la norma IEC 61131.

Va ser molt important per als sistemes de control distribuïts la incorporació dels blocs de funcions de control. Els blocs de funcions de control són blocs programables amb memòria. Disposa d'un bloc de dades assignades com memòria (bloc de dades d'instància). Els paràmetres que es transfereixen al FB, així com les variables estàtiques, es memoritzen. Les variables temporals es memoritzen a la pila de dades locals. Els FB són funcions predefinides que es poden utilitzar sempre que necessitem.

El que va comportar una millora notable en els DCS van ser la comunicació digital entre els controladors distribuïts, estacions i altres computadores. Els DCS necessiten que tota la instrumentació del procés i el sistema de control estiguin comunicats (senyals analògics dels sensors de camp cap als controladors per la traducció, i a través d'aquests senyals fer el control del procés a través dels actuadors). Actualment existeixen diferents protocols de comunicació que permeten comunicar dispositius amb senyals de 4-20 mA, així com busos de camp complexos com el Foundation Fieldbus, el Profibus, o el HART.

Tot aquest sistema de control distribuït a la indústria, ha evolucionat conjuntament amb les comunicacions industrials. Actualment es realitza un control/supervisió de manera remota, amb Internet com a eina d'implementació. En els orígens cada fabricant disposava del seu protocol de comunicació, aquest fet va originar una gran diversitat de protocols i, per tant, un problema de comunicació entre equips. Per solucionar-lo 5 empreses fabricants d'equips de control: Intellution, Opto-22, Fisher-Rosemount, Rockwell Software i Intuitive Software, conjuntament amb Microsoft van acordar treballar juntes per tal d'eliminar aquesta problemàtica a les comunicacions. Van definir les interfícies basades en OLE/COM (Object Linking and Embedding / Common Object Model), donant lloc al concepte de sistemes de control oberts: OPC (Ole for Process Control). Al mateix temps es va fundar la "OPC Foundation"<sup>1</sup> una organització que es dedica a la regularització dels estàndards de comunicació. OPC és un estàndard que regula un idioma comú per a l'intercanvi d'informació entre diferents tipus, gammes de dispositius i fabricants d'automatització.

La majoria de senyals I/O analògics (4..20 mA, 0..10 V, etc...) aporten informació de variables físiques com pressió, temperatura, cabal, etc... Destaquen per la seva funcionalitat per al control continu (llaços de control PID), control seqüencial, alarmes, i la possibilitat d'utilitzar algorismes de control complexes. Presenten un temps de cicle més alt que els PLCs, de l'ordre de dècimes de segon, per aquest motiu la seva utilització és viable per al control de processos químics on els temps de resposta de les variables a controlar són de l'ordre de minuts.

Els PIDs (Controlador Proporcional Integral Derivatiu) són un mecanisme de control per realimentació utilitzat en la gran majoria de controls industrials. S'utilitza per a la correcció de l'error d'una mesura, calcula l'error d'un valor mesurat i el valor desitjat.

L'algoritme consta de 3 paràmetres: el proporcional, l'integral, i el derivatiu. El proporcional serveix per corregir l'error actual, es calcula amb el producte de l'error per la constant proporcional ( $K_p$ ), per aconseguir que l'error en estat estacionari sigui el més pròxim a 0. Existeix un valor límit del valor de la constant proporcional a partir del qual el sistema pren un valor superior al desitjat, això es diu sobreoscil·lació, que mai pot sobrepassar del 30%. Com que l'acció proporcional no depèn dels temps, la millor manera de tractar els errors respecte la variació amb el temps, és incloure l'acció integral i la derivativa. L'acció integral permet corregir l'error acumulat i l'acció derivativa permet corregir els possibles errors futurs.

---

<sup>1</sup> <https://opcfoundation.org/>



## 1.2. JUSTIFICACIÓ DEL PROJECTE

El punt de partida del projecte és la disponibilitat de PLC d'última generació, gràcies a una donació realitzada per l'empresa B&R Automation. Aquests PLC s'han incorporat a quatre estacions de laboratori dissenyades a partir d'unes plantes ja disponibles de FESTO.

Les estacions originals eren estacions del tipus DCS, ja que cada una disposava del seu control PID per la regulació d'una variable, enllaçades mitjançant una estació distribuïdora governada per un PLC. Les estacions dissenyades es poden considerar estacions híbrides des del punt de vista del control, ja que integraran parts lògiques i parts de regulació de variables.

A l'hora de realitzar el treball, adquirirem uns coneixements sobre els elements de control i funcionament de cada estació. A més de conèixer la manera de treballar amb els PLCs subministrats per B&R i la manera de programar-los amb ladder. La normativa IEC 1131 publicada l'any 1993 per la Comissió Electrotècnica Internacional és l'estàndard internacional pels llenguatges de Controladors Lògics Programables. L'última revisió d'aquesta normativa és de l'any 2013 l'IEC 61131-3.

## 1.3. OBJECTIU

L'objectiu d'aquest projecte és el de desenvolupar, automatitzar i monitoritzar una estació de distribució d'aigua, en la qual a partir de la informació obtinguda dels sensors distribuïts per l'estació de treball es regulin variables com pressió, nivell i flux mitjançant els actuadors presents a l'estació de treball. Aquestes estacions seran controlades mitjançant els PLCs subministrats per B&R Automation.

L'objectiu principal s'assoleix amb els següents objectius específics:

- Cercar d'informació sobre els actuadors i sensors presents en l'estació de treball;
- Disseny i desenvolupament de l'estació de treball;
- Aprenentatge del llenguatge de programació dels PLCs emprats;
- Definició de les especificacions de funcionament de l'estació i la seva programació;
- Aprenentatge del funcionament la pantalla per a tasques de monitorització.



## 1.4. FINALITAT DEL PROJECTE

Aquest projecte té dues finalitats:

- Una seria per al sector industrial, que estaria enfocat en processos de tractaments d'aigua com depuradores, amb processos de distribució d'aigua, i definitivament en tot procés en el qual es vulgui tenir control de la circulació d'aigües.
- L'altra finalitat seria en el sector acadèmic, ja que les estacions serviran per realitzar pràctiques per a futurs alumnes de l'EPSEM, ja que tindran la possibilitat de realitzar diverses pràctiques de diferents complexitats i tipus de controls, amb els sensors i els actuadors que tenim. En aquest aspecte estaria enfocat per als alumnes d'enginyeria electrònica i automàtica. Per als químics estaria enfocat per als diferents tractaments físics de l'aigua a través del control i l'automatització.





## 2. ANTECEDENTS

Els antecedents en la realització del treball són dos, per una banda hi ha el TFG realitzat el curs 2017-2018 pel Jaume Sevitja i l'altra les estacions que han servit de base per al desenvolupament de la maqueta. Per a la realització del projecte s'han aprofitat cinc estacions dedicades a l'aprenentatge de l'automatització i control a nivell industrial.

L'empresa *B&R Industrial Automation GmbH* ens ha subministrat quatre PLCs d'última generació y seran els utilitzats per a l'automatització de les noves estacions dissenyades. Aquests permeten, a més a més de poder programar en LADDER, fer programació amb TEXT ESTRUCTURAT, que és on el món de la programació de PLCs actual s'està orientant. L'empresa va oferir un curs de 40 h al qual vaig assistir el Novembre de 2017.

### 2.1. Treballs fi de graus anteriors

El treball fi de grau realitzat per un company de la universitat, en Jaume Servitja, el curs 2017-2018 titulat **Desenvolupament d'eines didàctiques per a l'aprenentatge de control de processos industrials** va tenir com a objectiu estudiar la possibilitat de realitzar unes pràctiques acadèmiques per l'estudi del control automàtic en l'àmbit de l'indústria química. Per tal de realitzar el guió de pràctiques es plantegen, s'apliquen i s'avaluen diverses tècniques de modelització de sistemes i sintonia d'estructures de control PID. Aquest treball ha servit de suport en el model i regulació de nivell de la nova estació.

### 2.2. Les nostres estacions

L'estructura completa del sistema en xarxa de FESTO consta de 4 estacions que permeten fer la regulació de nivell, de cabal, de pressió i temperatura, i una estació de distribució que permet connectar els dipòsits principals de cada estació de regulació entre ells.

a) Estació de control de nivell

La funció principal del sistema de control de nivell (Figura 2.) és la regulació del nivell de líquid en el tanc superior de reserva, alimentat a través d'una bomba la qual es troba connectada a un altre tanc la funció bàsica del qual és l'emmagatzematge del líquid. El control de nivell es fa amb el sensor analògic de nivell.

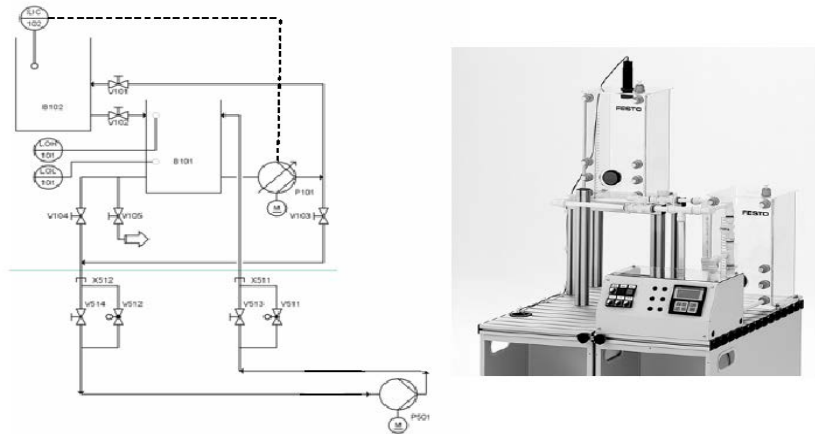


Figura 2. Diagrama de flux i foto del procés de control de nivell.

**b) Estació de control de cabal**

La funció principal del sistema de control de cabal (Figura 3. ) és la regulació del cabal que circula per un tub. Es disposa d'un sensor optoelectrònic i la regulació es pot fer variant el cabal mitjançant una vàlvula proporcional o una bomba motoritzada.

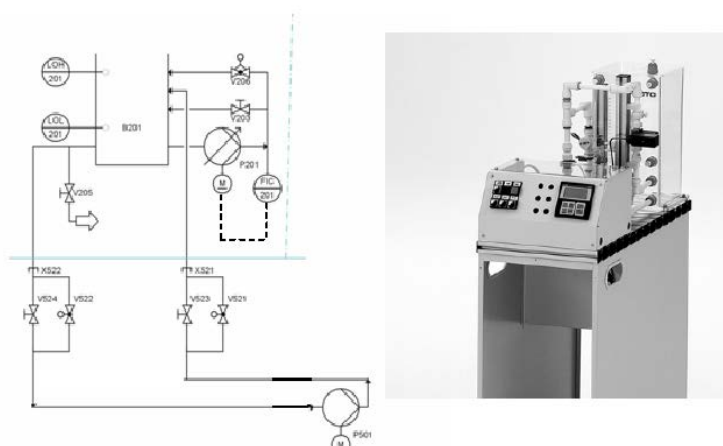


Figura 3. Diagrama de flux i foto del procés de control de cabal.



### c) Estació de control de pressió

La funció principal del sistema de control de pressió (Figura 4.) és la regulació de la pressió que aconseguim al vas d'expansió. Es disposa d'un sensor de pressió i la regulació es pot fer variant la pressió mitjançant una bomba motoritzada.

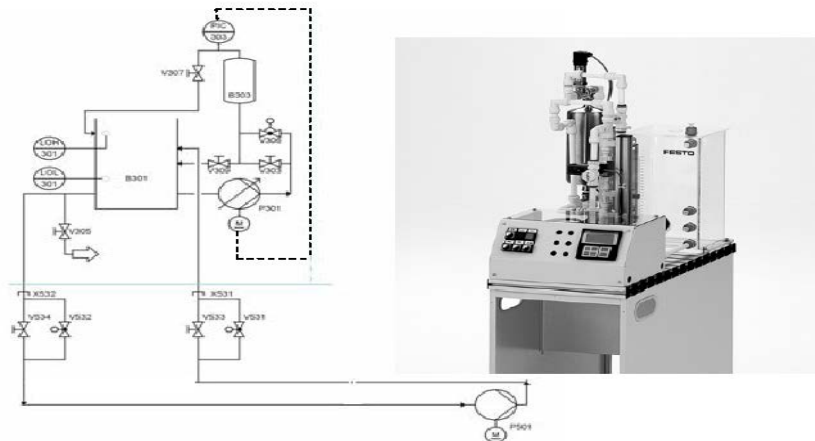


Figura 4. Diagrama de flux i foto del procés de control de pressió

### d) Estació de control de temperatura

La funció principal d'aquesta estació és el control de temperatura (Figura 5) del líquid que hi ha dins del dipòsit, dins del qual hi ha una resistència que escalfa l'aigua realitzant així el control de temperatura i ajustar-la a la temperatura desitjada. El control s'obté amb un sensor analògic de temperatura.

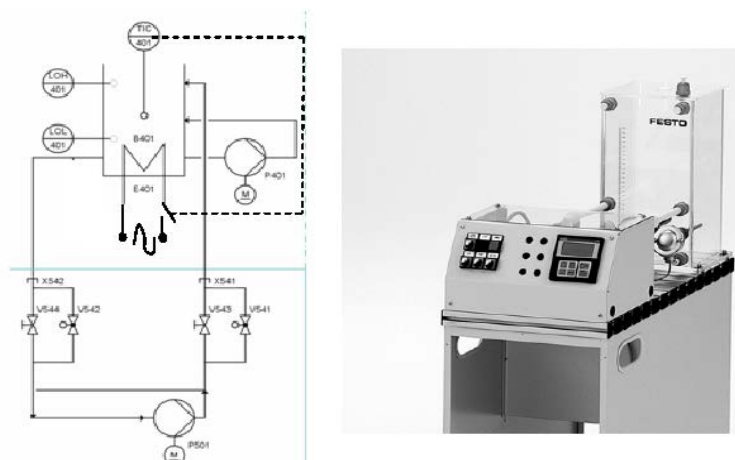


Figura 5. Esquema i foto del procés de control de temperatura.



En tots els processos a més a més dels elements de control: sensors, motobomba i vàlvules, hi ha 4 interruptors: on/off del procés (POWER), on/off de la vàlvula motoritzada (VALVE), un interruptor sense acció (EXTERNAL FUNCTION) i 2 indicadors de control per visualitzar que el procés està engegat (llum verda) i un altre d'alarma (llum blava). També es disposava de 4 PIDs i un PLC de la casa OMRON.

## 3. DISSENY L'ESTACIÓ DE TREBALL

### 3.1. Descripció general de la maqueta

L'estació de treball dissenyada consta de 2 dipòsits ubicats a diferents alçades i connectats per tubs que permeten el traspàs d'aigua entre ells, i un vas de pressurització. Per fer circular l'aigua es disposa d'una motobomba de cabal variable. En els tubs hi ha una sèrie d'electrovàlvules amb les quals podem fixar el recorregut de l'aigua per l'estació, com a mesura de seguretat i per introduir perturbacions en el sistema s'han incorporat vàlvules d'accionament manual en paral·lel a les electrovàlvules.

Un conjunt de detectors lògics instal·lats en els dipòsits ens permeten conèixer l'estat dels dipòsits (ple, buit, mig ple). També es disposa d'un sensor de nivell analògic que ens permet conèixer el nivell d'aigua d'un dels dipòsits de l'estació de treball. L'estat del tanc pressuritzat ens els dona un sensor de pressió.

Tot el control es realitza amb el PLC de B&R d'última generació. Aquest PLC permet accedir tant a senyals lògics com analògics, per tant ens permet controlar les electrovàlvules, la motobomba i accedir a la informació proporcionada pels sensors.

Es disposa d'un teclat d'operació amb el qual podrem entrar les ordres d'operació i rebre informació de l'estat general del sistema. També s'ha previst complementar aquest teclat amb una pantalla interactiva.

### 3.2. Esquema de la maqueta

La Figura 6, ens mostra l'esquema de la estació 1. Com a sensors o transductors disposem de:

- 5 sensors de nivell tot o res (lògics), dos estan situats al tanc superior, ALT, i s'identifiquen com LT1.1 i LT1.2, i els altres tres estan situats al tanc inferior, BAIX, i s'anomenen LT1.3, LT1.4, LT1.5;
- un sensor de nivell analògic, LT1.6, situat al tanc ALT, amb el qual es podrà realitzar el control del nivell.
- Un sensor de pressió, PT1.1, situat al vas d'expansió, que ens permetrà realitzar el control de pressió del líquid dins del vas;
- I, un sensor de flux, FT1.1, que ens permet conèixer el flux d'aigua que circula pel tub que comunica el dipòsit BAIX amb ALT.

En quant a actuadors, disposem de:

- Una motobomba, P1.1;
- 4 electrovàlvules normalment tancades distribuïdes al llarg de l'estació, V1.1, V1.2, V1.3 i V1.4.
- Una electrovàlvula motoritzada, V1.5An, que permet regular el pas d'aigua entre els dipòsits ALT i BAIX.
- 5 vàlvules manuals en paral·lel amb les electrovàlvules.

Com és pot observar a la Figura 6 l'aigua pot anar des del tanc BAIX al tanc ALT activant la motobomba P1.1 i obrint electrovàlvula V1.4 i la podem retornar al tanc BAIX obrint l'electrovàlvula V1.5 An. També podem fer circular l'aigua des del tanc BAIX fins al vas d'expansió activant la motobomba i obrint les electrovàlvules V1.1 i V1.2. Podem també fer circular l'aigua pel tanc inferior obrint les electrovàlvules V1.1 i la V1.3 simultàniament.

## ESTACIÓ 1

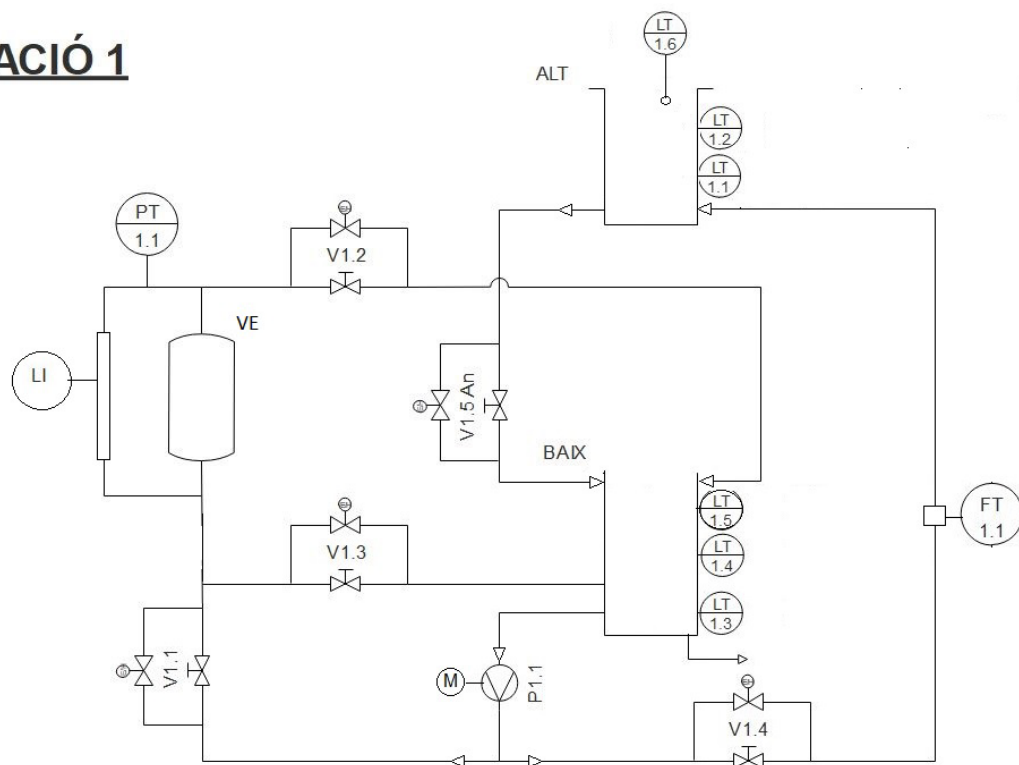


Figura 6. Esquema de l'estació 1.



Figura 7. Imatge de l'estació 1.

A la Figura 7, es mostra la imatge de l'estació 1 el dia de la firma del contracte amb la empresa B&R que ens ha proporcionat els PLC amb els quals controlem les estacions.

### 3.3. Descripció dels sensors i actuadors

En aquesta secció es descriuen les característiques bàsiques dels sensors i dels actuadors utilitzats. Aquesta informació es pot complementar amb la informació dels datasheet descrits en els manuals de les maquetes de FESTO.

#### 3.3.1. Sensors de nivell

Com s'ha vist en l'esquema de la maqueta (Figura 6) l'estació disposa de dos tipus de sensors de nivell: 5 sensors tot o res i un sensor analògic.

Els 5 sensors tot o res son de tipus capacitius, amb connexió PNP normalment obert. Poden treballar entre un rang de tensió de 10 VDC a 55 VDC. Disposen d'ajust de sensibilitat de 2 a 8 mm amb una histèresi del 3 al 15%. La màxima freqüència de treball és de 300 Hz. El fabricant diu que la temperatura de treball ha de ser de 20°C fins a 70°C.



I, tenen un nivell de protecció IP-65. El seu funcionament es basa en la variació de la capacitat del condensador que hi ha en el circuit intern RC. La capacitat augmenta quan s'aproxima el material al sensor, i aquesta variació depèn del material, de la distància i de la constant dielèctrica del material. Com a indicadors de funcionament hi ha un LED verd, que indica que està alimentat, i un altre LED groc/taronja, que s'activa quan es detecta el material que en el nostre cas és aigua.

Per al control de nivell disposarem d'un sensor d'ultrasons analògic situat al tanc superior que dona una sortida de 4 a 20 mA, amb un rang d'alimentació de 10 a 30 VDC. Té una resolució màxima de 0,4 mm, en un rang de 50 fins a 300 mm. El fabricant ens diu que pot treballar en entorns de temperatura de -25°C fins a 70°C. El nivell de protecció és una IP-67. Com més distància tingui respecte el material major és la diferència entre +UB i -UB. +UB i -UB són les entrades de tensió amb els quals calculant la diferència surt la distància. Com més gran és la diferència, ens retornarà un corrent més gran. El led GROC ens indica que està connectat i el LED vermell que ens indica que s'ha produït algun tipus d'error.

### 3.3.2. Sensor de pressió

Per al control de pressió disposem d'un sensor de pressió piezorresistiu analògic alimentat a 24 VDC que ens retorna un valor de 0 VDC a 10 VDC proporcional a la pressió.. El rang de mesura va de 0 mBar fins a 100 mBar. Pot treballar de 0°C fins a 65°C segons el fabricant.

### 3.3.3. Sensor de cabal

Per al control de cabal disposarem d'un sensor optoelectrònic que pot alimentar-se de 5 VDC a 12 VDC amb un consum de 6 mA a 33 mA. El seu funcionament es basa en unes pales que giren a una velocitat proporcional a la quantitat de líquid que hi passa. Contra més cabal més ràpid giren les pales i per tant mesura més cabal. Gràcies al díode i al fototransistor es proporciona un senyal de polsos de 5 VDC a 12 VDC amb un rang de freqüència de 13 Hz fins a 1200 Hz. El rang de mesura és de 0,5 fins a 15 l/min, i la màxima pressió a la qual pot treballar és a 6 bar a 80°C. El rang de temperatures de treball és de 0°C fins a 65°C. Solament pot treballar en un sentit de circulació, mai retornarà un senyal negatiu.

### 3.3.4. Electrovàlvules

Es disposa de 4 electrovàlvules o vàlvules solenoïdals que s'utilitzen per controlar el flux (ON-OFF) de l'aigua. Estan alimentades a 24 VDC i tenen un consum de 8 W. El seu funcionament es basa en el fet que quan li arriba senyal s'excita la bobina i obra o tanca el pas de líquid amb l'èmbol. Poden treballar de 0 a 0,5 bar i temperatures de 0 a 65°C.

Hi ha també una electrovàlvula proporcional analògica, alimentada també a 24 VDC i pot treballar en el mateix rang de pressió i temperatura que les vàlvules solenoïdals. La seva obertura es controla amb un senyal de rang entre 0 a 10 VDC.

### 3.3.5. Motobomba

Per fer circular l'aigua per l'estació s'ha instal·lat una bomba centrífuga que serà la que aporti més o menys cabal a l'aigua. Està alimentada a 24 VDC, i un consum de 0,5 a 0,9 A, i depenent del senyal de control girarà més o menys ràpid. El cabal màxim que ens pot aportar és de 10 l/min.

## 3.4. Teclat d'operació



Figura 8. Teclat d'operació.

Com podem veure a la Figura 8, el teclat d'operació disposa de:

- Un selector ON/OFF per alimentar dels mòduls d'entrades i sortides del PLC i alimentar els actuadors i sensors.





- Un pulsador Start que el podrem fer servir per programa com a botó de marxa del batch.
- Un selector A/M (Automàtic/Manual) per seleccionar si el procés el volem fer de manera automàtica o en el cas de treballar manualment amb cada element del circuit.
- Un pulsador STOP que per programa es podrà fer servir per parar el procés allà o que es vulgui.
- Un interruptor d'ATURADA D'EMERGÈNCIA que talla l'alimentació de la motobomba.
- Un LED ALARMA que podrem programar perquè s'encengui quan hi hagi un error o fallada.
- Un LED POWER que per programa podrem fer que s'encengui quan el sistema està en marxa.





## 4. SISTEMA DE CONTROL I AUTOMATITZACIÓ

### 4.1. X20 cp 1583 / X20 CP 1381

#### 4.1.1. CPU X20 CP 1381

La CPU del PLC en el nostre cas és CPU X20 CP 1381. Disposa de 128 MB DDR3 de RAM i de 16 kB de FRAM. S'alimenta a 24V i consumeix un màxim de 1 A d'entrada de corrent. Té un fusible integrat el qual no es pot canviar. La CPU consta de tres mòduls X1, X2 i X3 (veure Figura 9). En les CPU, hi ha la font d'alimentació integrada. Es pot connectar l'alimentació per les CPU, X2X link i la font d'alimentació interna d'entrades i sortides.

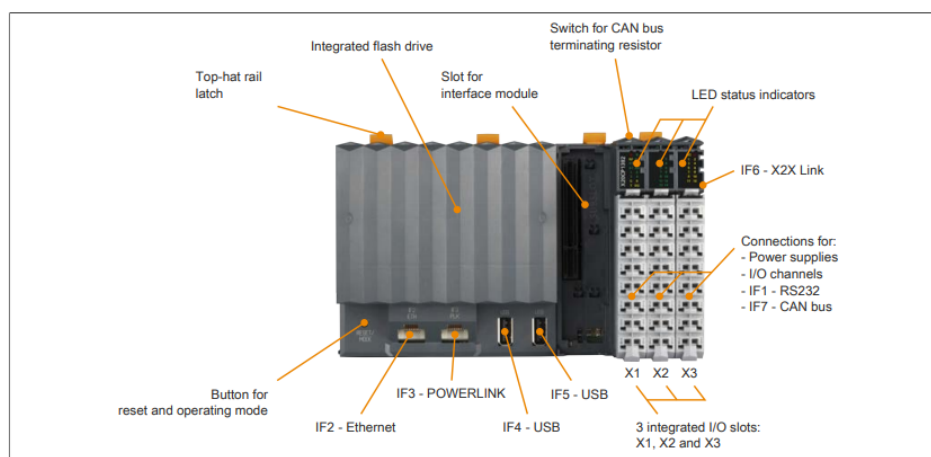


Figura 9. X20CP1381 amb els mòduls.

#### a) Mòdul X1:

En el mòdul X1, Figura 10, hi trobem la CPU amb les entrades analògiques, que les podem utilitzar per tensió o per corrent ("U" per tensió de -10V a 10V i "I" per corrent de 0 20mA o de 4 a 20 mA), també disposa de 4 entrades digitals (DI). A la Taula 1 podem veure totes les possibles connexions d'entrades que podem realitzar en aquest mòdul.

<sup>2</sup><https://www.br-automation.com/es/productos/sistemas-de-control/sistema-x20/cpus-x20/x20cp1381/#downloads>

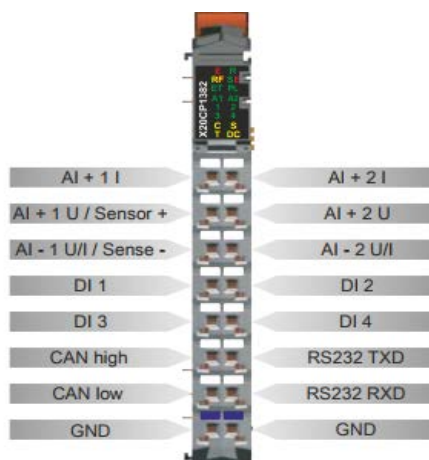


Figura 10. Mòdul X1

Taula 1. Connexions X20CP1381

#### Digital inputs/outputs

Connection	Terminal connection	Channel	Description
X1	14	DI 1	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	24	DI 2	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	15	DI 3	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	25	DI 4	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter

#### Analog inputs

Connection	Terminal connection	Channel	Description
X1	11, 12, 13	AI 1	$\pm 10 V / 0$ to 20 mA or 4 to 20 mA, 12-bit, 1 ms
	21, 22, 23	AI 2	$\pm 10 V / 0$ to 20 mA or 4 to 20 mA, 12-bit, 1 ms

Analog input 1 can also be used for PT1000 resistance temperature measurement.

Connection	Terminal connection	Channel	Description
X1	11, 12, 13	AI 1	PT1000 resistance temperature measurement Measurement takes place using analog input A1.

#### b) El mòdul X2:

El mòdul X2 disposa de 10 entrades digitals de 24V ,4 de les quals són de comptatge ràpid ( $2 \mu s$ ) per connectar-hi encòders. A la Figura 11 podem veure físicament el mòdul X2 on les entrades normals digitals van des de la DI 1 fins la DI 10, i les entrades de comptatge ràpid van des de la DI 11 (high-speed) fins la DI 14 (high speed). I, a la Taula 2, podem veure totes les possibles connexions i característiques d'aquestes.

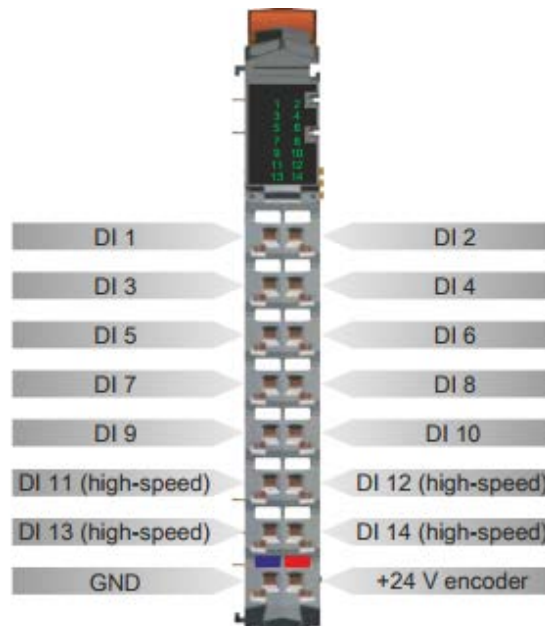


Figura 11. Mòdul X2

Taula 2. Connexions mòdul X2

X2	11	DI 1	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	21	DI 2	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	...	...	...
	25	DI 10	24 VDC, sink, $\leq 200 \mu s$ , configurable software filter
	16	DI 11	24 VDC, sink, $\leq 2 \mu s$ , configurable software filter
	26	DI 12	24 VDC, sink, $\leq 2 \mu s$ , configurable software filter
	17	DI 13	24 VDC, sink, $\leq 2 \mu s$ , configurable software filter
	27	DI 14	24 VDC, sink, $\leq 2 \mu s$ , configurable software filter

### c) El Mòdul X3:

El mòdul X3 disposa de 4 sortides digitals de 24V- 0,5 A, 4 entrades/sortides digitals de 24V-0,5 A i 4 sortides de comptatge ràpid (2  $\mu s$ ), de 24V- 0,2 A. Les d'impulsos es poden fer servir com a sortides PWM. A la Figura 12 podem veure físicament el mòdul X3 amb les seves sortides. I a la Taula 3 les sortides i les seves característiques.

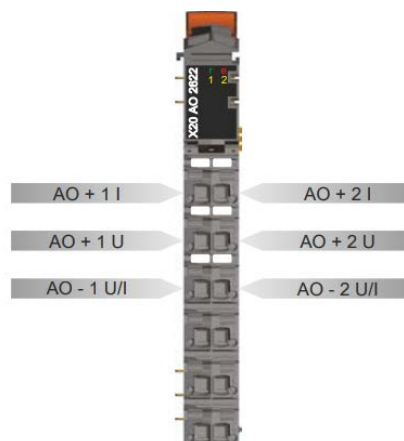


Figura 12. Mòdul X3

Taula 3. Connexions mòdul X3

X3	11	DO 1	24 VDC, 0.5 A, source, <300 µs
	21	DO 2	24 VDC, 0.5 A, source, <300 µs
	12	DO 3	24 VDC, 0.5 A, source, <300 µs
	22	DO 4	24 VDC, 0.5 A, source, <300 µs
	13	DI 5 / DO 5	DI: 24 VDC, sink, ≤200 µs, configurable software filter DO: 24 VDC, 0.5 A, source, <300 µs
	23	DI 6 / DO 6	DI: 24 VDC, sink, ≤200 µs, configurable software filter DO: 24 VDC, 0.5 A, source, <300 µs
	14	DI 7 / DO 7	DI: 24 VDC, sink, ≤200 µs, configurable software filter DO: 24 VDC, 0.5 A, source, <300 µs
	24	DI 8 / DO 8	DI: 24 VDC, sink, ≤200 µs, configurable software filter DO: 24 VDC, 0.5 A, source, <300 µs
	15	DO 9	24 VDC, 0.2 A, push-pull, <3 µs
	25	DO 10	24 VDC, 0.2 A, push-pull, <3 µs
	16	DO 11	24 VDC, 0.2 A, push-pull, <3 µs
	26	DO 12	24 VDC, 0.2 A, push-pull, <3 µs

#### d) El mòdul d'expansió X4:

En el nostre cas disposem d'un 4rt mòdul d'expansió el qual té dues sortides analògiques que treballen amb tensió o amb corrent, com es pot veure a la Figura 13. La potència que dona al PLC és de 2 W. El cable de les entrades/sortides de comptatge ràpid pot tenir una llargada màxima de 20 metres. El rellotge de la CPU treballa a 200 MHz, i la tasca més ràpida es realitza cada 2 ms.

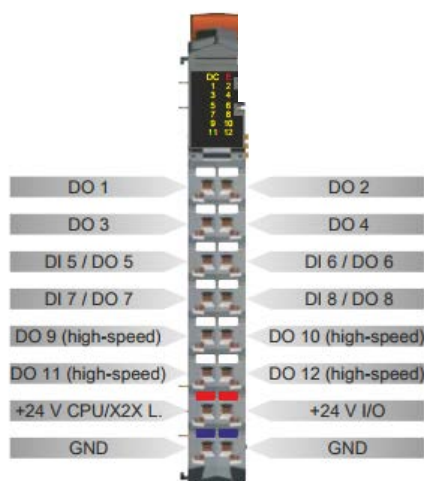


Figura 13 . Mòdul X4

Tots els datasheets del nostre PLC els podem trobar a la pàgina web: <https://www.br-automation.com/en/products/control-systems/x20-system/x20-cpus/x20cp1381/>.



## 4.2. Connexions amb els sensors i actuadors

#### 4.2.1. Connexió dels sensors de nivell digitals

Com podem veure a la Taula 4, els sensors de nivell digitals tenen 3 cables: el negre, que és el de senyal, el connectarem directament a l'entrada digital del mòdul X2 que li pertoca, el cable blau, que és el de 0 V, el connectarem a la placa de connexions que hem utilitzat i el cable vermell el connectarem als 24 V de la placa de connexió. Els sensors LT4 i LT5 s'alimenten a la placa de connexió de l'estació 2. A la Figura 14 podem veure l'esquema elèctric dels sensors de nivell digitals amb la placa de connexions que utilitzem, i a la Figura 15 on van connectats del PLC.

#### Taula 4.Taula connexions sensors digitals

MODUL X2			
ENTRADES	CONNEXIÓ AL MODUL Y A LA PLACA DE CONNEXIONS		
SENSOR DE NIVELL DIGITAL LT1.1	DI 1 (cable negre,senyal)	Cable marró (24V) posició 20	Cable gris (0 V) posició 34
SENSOR DE NIVELL DIGITAL LT1.2	DI 2 (cable negre,senyal)	Cable marró (24V) posició 21	Cable gris (0 V) posició 36
SENSOR DE NIVELL DIGITAL LT1.3	DI 3 (cable negre,senyal)	Cable marró (24V) posició 16	Cable gris (0 V) posició 38
SENSOR DE NIVELL DIGITAL LT1.4	DI 4 (cable negre,senyal)	Cable marró (24V)	Cable gris (0 V)
SENSOR DE NIVELL DIGITAL LT1.5	DI 5 (cable negre,senyal)	Cable marró (24V)	Cable gris (0 V)

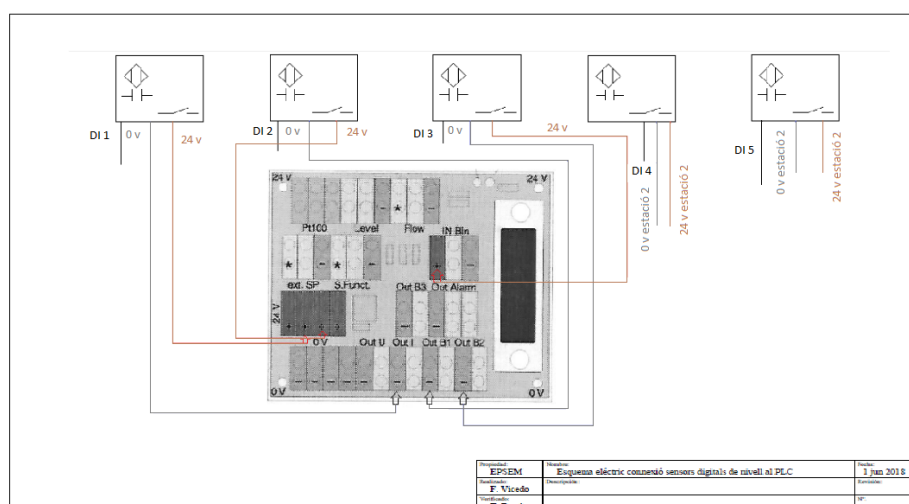


Figura 14. Esquema elèctric de les connexions dels sensors digitals de nivell

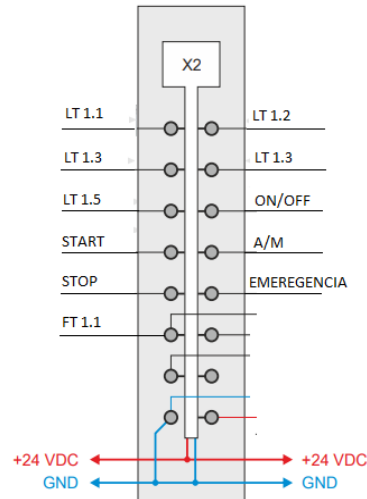


Figura 15. Connexions al PLC

#### 4.2.2. Connexió de les electrovàlvules digitals

Les electrovàlvules digitals són de 2 cables negres, un el connectarem directament a les sortides digitals del mòdul X3, i l'altre als 0 v de la placa de connexió com es pot veure a les imatges. A la Figura 17 tenim l'esquema elèctric de les electrovàlvules i a la Figura 16 com van connectades del PLC. A la Taula 5 podem veure la taula de connexions.

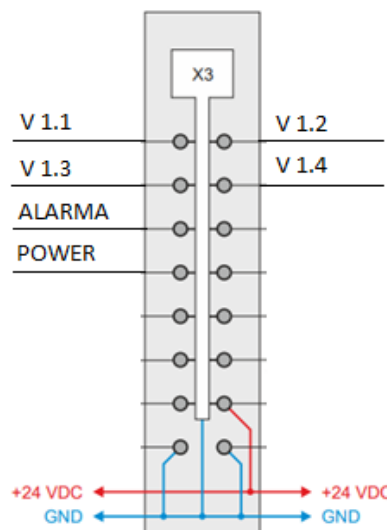


Figura 16. Esquema elèctric electrovàlvules digitals

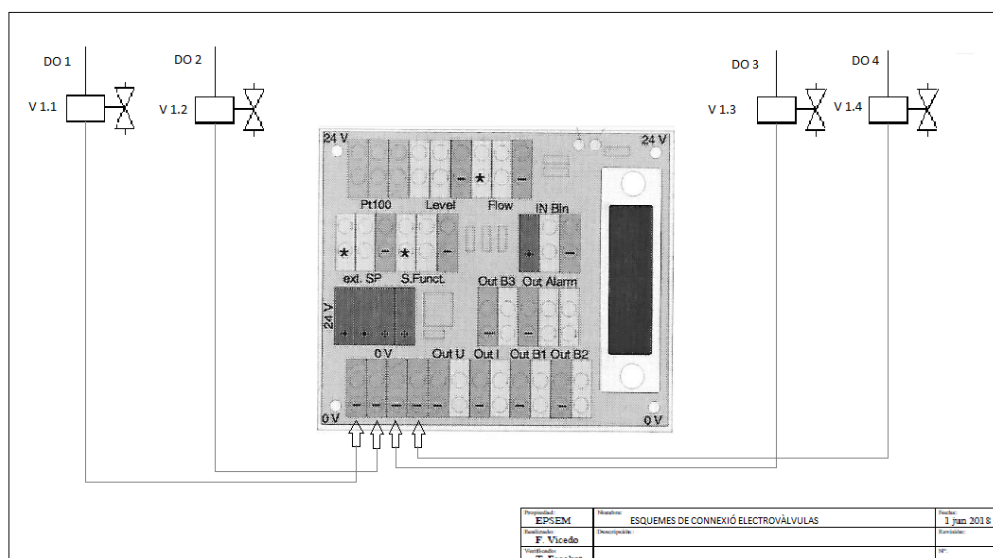


Figura 17. Esquema de connexions de les electrovàlvules al PLC

Taula 5. Connexions electrovàlvules

MODUL X3		
SORTIDES	CONNEXIÓ AL MODUL Y A LA PLACA DE CONNEXIONS	
ELECTROVÀLVULA DIGITAL V1.1	DO 1 negre	(0V) posició 28 cable gris
ELECTROVÀLVULA DIGITAL V1.2	DO 2 negre	(0V) posició 29 cable gris
ELECTROVÀLVULA DIGITAL V1.3	DO 3 negre	(0V) posició 30 cable gris
ELECTROVÀLVULA DIGITAL V1.4	DO 4 negre	(0V) posició 31 cable gris

### 4.2.3. Connexió motobomba i electrovàlvula V 1.5 (AN)

La motobomba té dos cables, un blau que va connectat a OUT(-) del variador i un cable marró que va a OUT(+). L'entrada del variador van, IN (+) a la sortida del PLC AO +2U i la sortida IN(-) va connectat a la sortida del PLC AO -2U. El variador va alimentat a 24 V i a 0 V. L'electrovàlvula V 1.5 analògica té tres cables: el negre, que és el de senyal, el connectarem a la sortida AO +1U, el cable marró, que el connectarem a 24 V a través de placa de connexió, i el cable blau, que el connectarem a la sortida del PLC AO -1U. A la Figura 18 tenim l'esquema elèctric de la motobomba i de l'electrovàlvula analògica V1.5. Com es pot observar la motobomba va connectada al variador, i d'aquest cap a la placa de connexions. A la Figura 19 veiem la connexió al PLC, i a la Taula 6 veiem la taula de connexions.

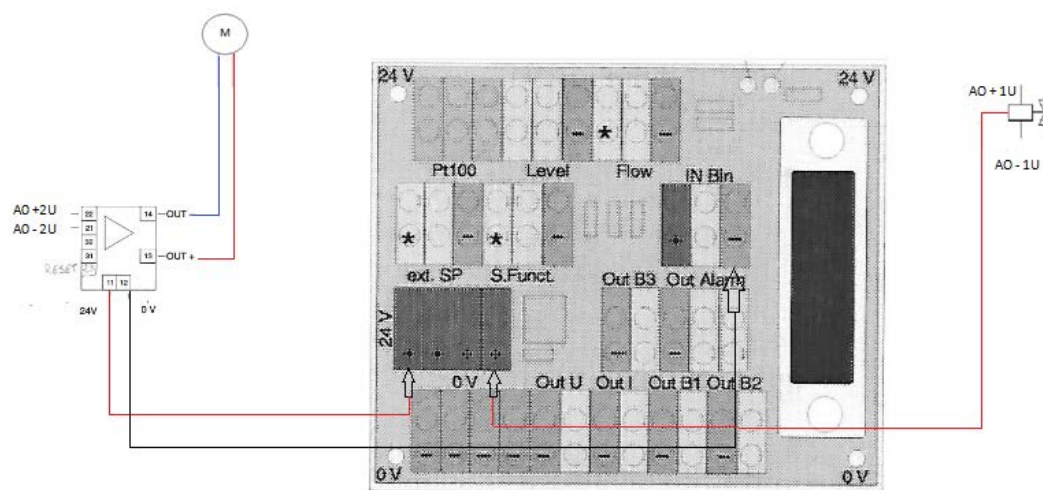


Figura 18. Esquema elèctric de la motobomba i la V1.5

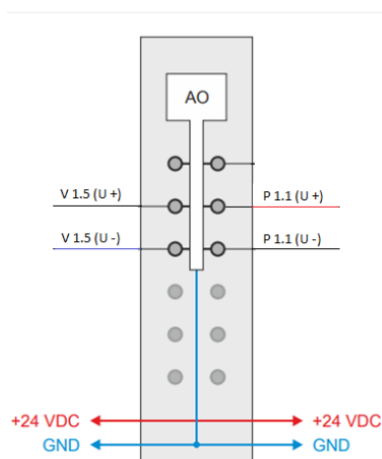


Figura 19. Connexions de la motobomba i l'electrovàlvula al PLC

Taula 6. Connexions motobomba i electrovàlvula analògica

MODUL X3 X20AO2622			
SORTIDES	CONNEIXIÓ AL MODUL Y A LA PLACA DE CONNEXIONS		
ELECTROVALVULA ANALOGICA V1.5	AO + 1U (Cable negre)	AO - 1 U/I (Cable blau)	24 V posició 22
MOTOBOMBA ANALOGICA P1.1	AO + 2U (Cable marró)	AO - 2 U/I (Cable blau)	



#### 4.2.4. Connexió sensors analògics

El sensor d'ultrasons de nivell dona una sortida de corrent de 4-20 mA. Com que volem treballar amb tensió farem servir un convertidor per passar el senyal de corrent a tensió.

El sensor de cabal l'alimentarem a la placa i el connectarem a l'entrada digital de comptatge ràpid DI 11 (high-speed). I el sensor de pressió el connectarem al AI +/-1U. A la Figura 20 tenim els esquemes elèctrics dels sensors analògics de cabal, pressió i nivell. A la Figura 21 es poden veure les connexions dels sensors al PLC, i a la Taula 7 la taula de connexions dels sensors.

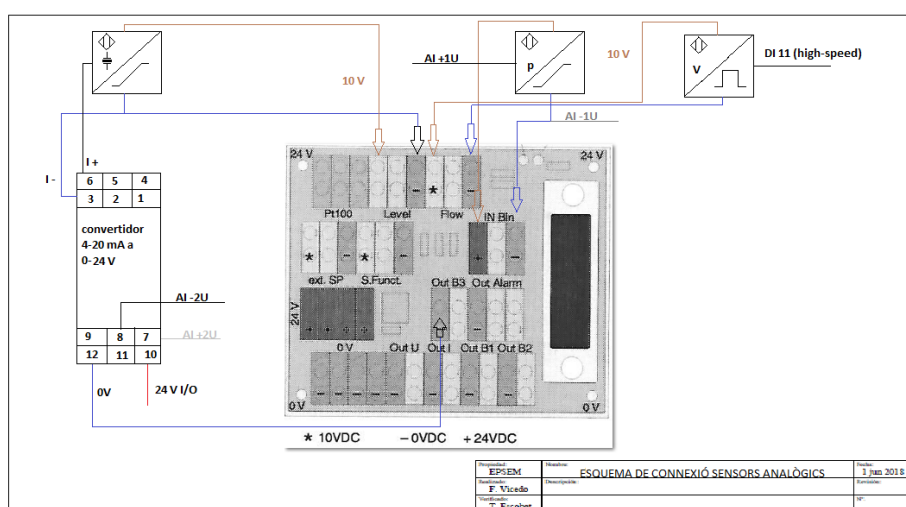


Figura 20. Esquemes elèctrics sensors analògics

Taula 7. Connexions sensors analògics

MODUL X1 CPU X20 CP 1381			
ENTRADES ANALÒGIQUES	CONNEXIÓ AL MODUL Y A LA PLACA DE CONNEXIONS		
SENSOR PRESSIÓ PT 1.1	AI + 1U (Cable negre)	AI - 1U/I (Cable blau)	Cable marró 24V
SENSOR DE NIVELL ANALÒGIC LT1.6	A2 + 2U (Cable negre)	AI - 2U/I (Cable blau)	Cable marró 24V
SENSOR DE CABAL FT 1.1	DI 11 (Cable negre)	0 v (posició 6)	Cable marró posició 4, 24V

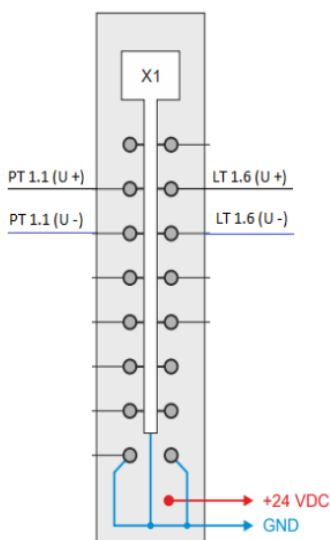


Figura 21. Connexions al PLC

#### 4.2.5. Connexió botonera, leds, i adquisició de dades

A la botonera disposem:

- D'un interruptor ON/OFF que és els que ens tallarà o donarà alimentació a les I/O del PLC.



- Un pulsador START que per programa el podrem utilitzar com a tal. Un selector automàtic-manual.
- Un interruptor d'emergència que ens donarà o tallarà l'alimentació de la bomba. I un pulsador d'STOP que per programa l'utilitzarem com a tal.
- 2 LEDs un d'ALARMA i un de POWER.

A la Figura 22 trobarem l'esquema elèctric de tots els pulsadors, interruptors i LEDs. A la Figura 15 podem veure on van connectats els pulsadors/interruptors al PLC i a la 17 els LEDs d'alarma i POWER. A la Figura 22 podem veure l'esquema de la botonera i a la Figura 23 la d'adquisició de dades.

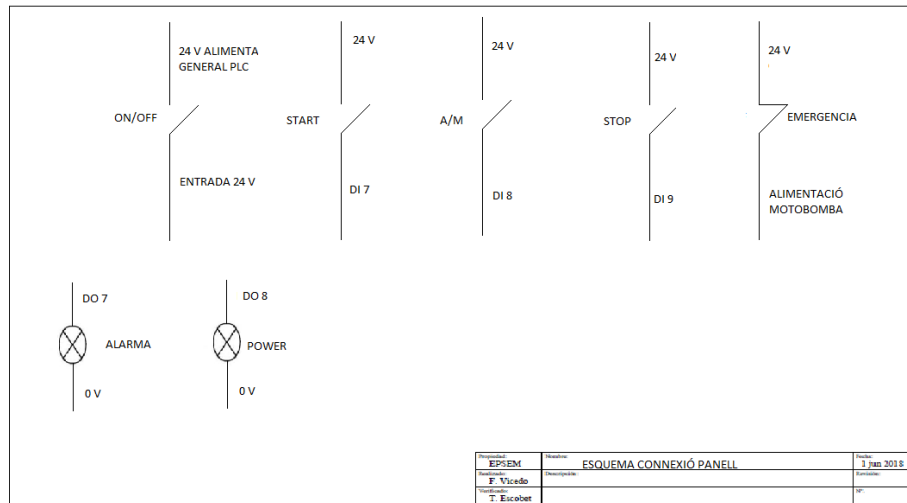


Figura 22. Esquema elèctric de la botonera

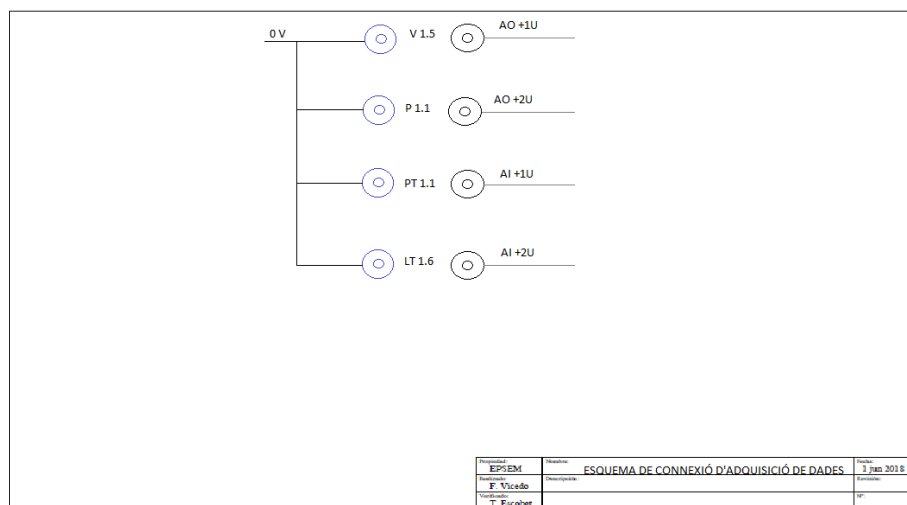


Figura 23. Connexions adquisició de dades



## 5. ENTORN DE PROGRAMACIÓ

### 5.1. Inicialització a l'Automation Studio

L'entorn de programació del PLC utilitzat s'anomena "Automation Studio". Al ser un entorn de programació força diferent dels utilitzats fins al moment, s'ha cregut necessari fer-ne una explicació detallada

El primer que haurem de fer és instal·lar-nos l'[Automation Studio](#). Un cop tinguem el programa instal·lat crearem un projecte nou on realitzarem els nostres programes. Obrirem l'Automation Studio i clicarem a "new project", com podem veure a la Figura 24.

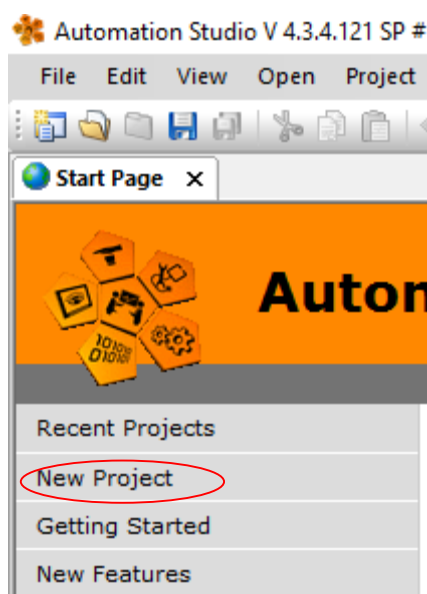
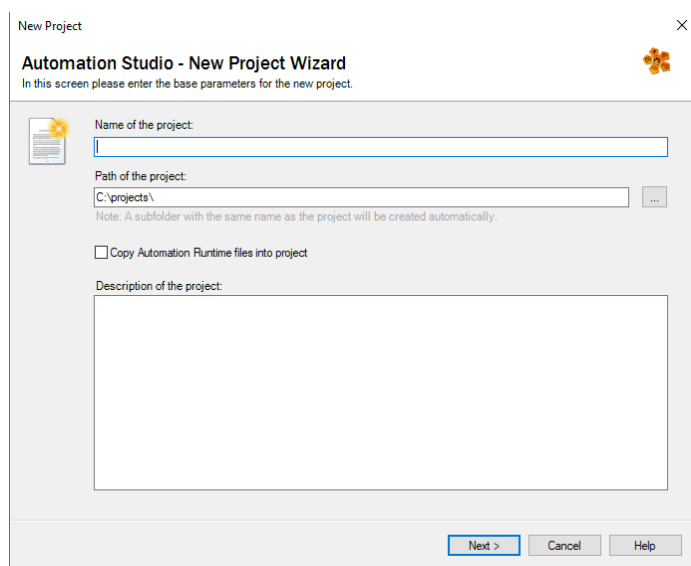


Figura 24. Imatge new project

Cliquem a "An empty project", i se'ns obrirà una finestra on posarem el nom que vulguem al projecte i la carpeta on volem que es guardi, Figura 25:

---

<sup>3</sup><https://br-automation.com/es/descargar/#categories=software/automation-studio/automation-studio-44>



New Project

**Automation Studio - New Project Wizard**

In this screen please enter the base parameters for the new project.

Name of the project:

Path of the project:

C:\projects\

Note: A subfolder with the same name as the project will be created automatically.

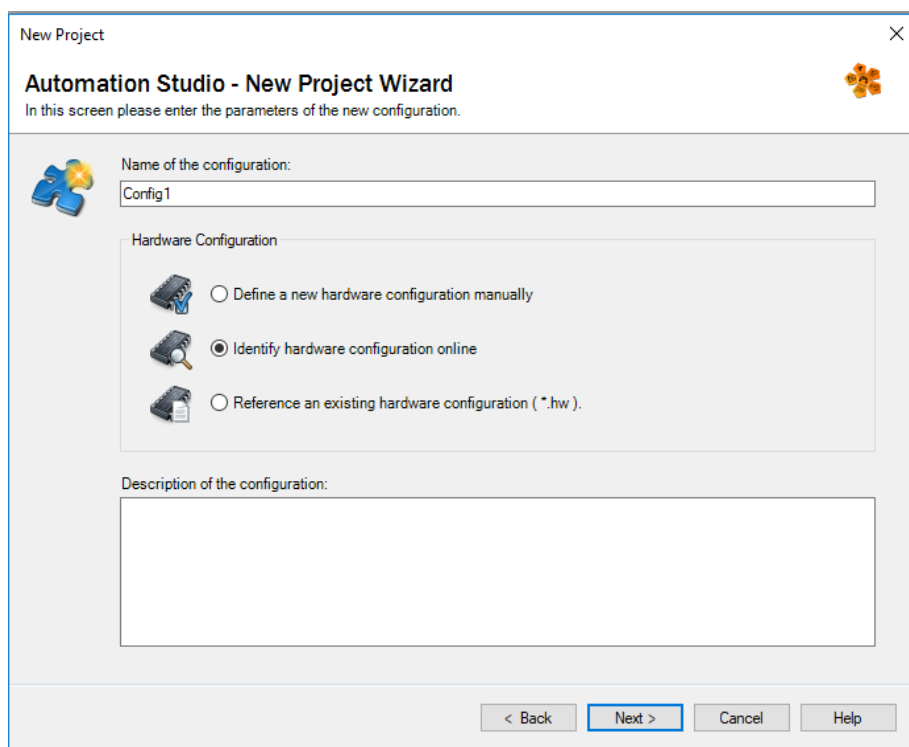
☐ Copy Automation Runtime files into project

Description of the project:

Next > Cancel Help

Figura 25. Imatge nom projecte

Un cop nomenat al nostre projecte haurem de definir manualment (o directament connectat amb el PLC), el tipus de PLC que volem programar, Figura 26:



New Project

**Automation Studio - New Project Wizard**

In this screen please enter the parameters of the new configuration.

Name of the configuration:

Config1

Hardware Configuration

☐ Define a new hardware configuration manually

☒ Identify hardware configuration online

☐ Reference an existing hardware configuration ( \*.hw ).

Description of the configuration:

< Back Next > Cancel Help

Figura 26. Identificació del PLC

Un cop ja tinguem el PLC seleccionat se'ns obrirà el projecte amb el PLC ja seleccionat, Figura 27. Se'ns obrirà automàticament la part de "Logical View" que és on incorporarem els programes, variables globals i locals, etc...

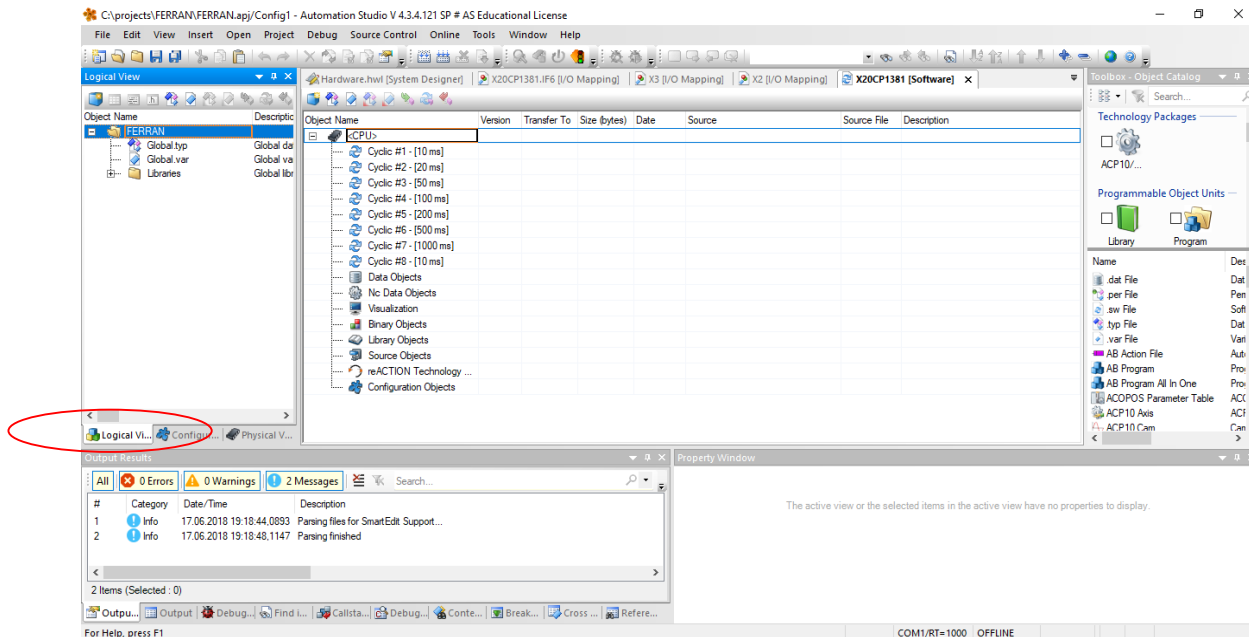


Figura 27. Finestra del nostre projecte

En aquest punt ja podem començar la programació del nostre projecte.

El que farem seguidament serà incorporar un programa decidint prèviament el llenguatge de programació a utilitzar, que en el nostre cas serà el Ladder, però recordem que el B&R té moltes altres opcions

Anirem a la finestra de dalt a la dreta anomenada Toolbox- Object Catalog i seleccionarem la finestra "Program", i se'ns obrirà una altra finestra amb tots els tipus de llenguatge. Seleccionarem LD Program (Ladder), com es veu a la Figura 28.

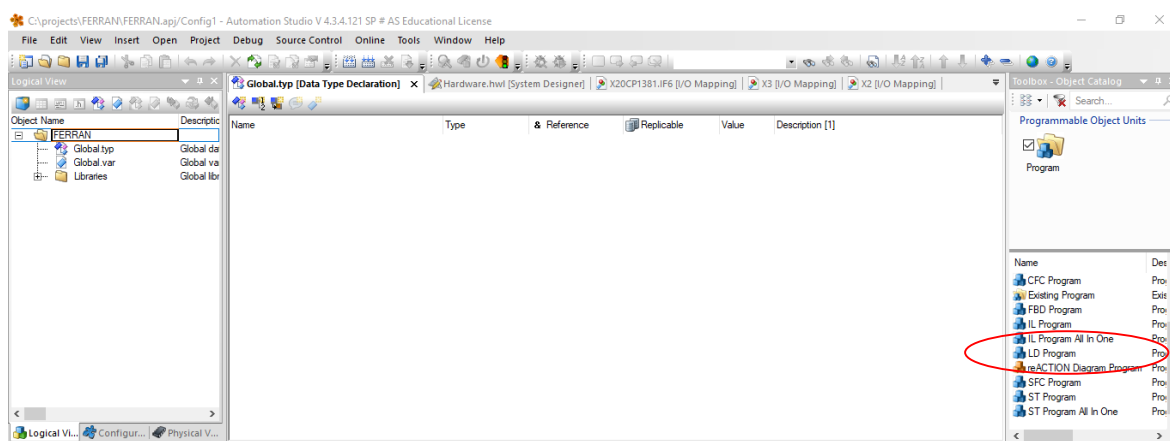


Figura 28. Imatge de selecció de programa

Ja haurem incorporat el nostre programa al nostre projecte. Dins del nostre programa tenim el “Cyclic” que serà el nostre programa principal, el “Init” que és el programa que només s’executarà el primer cicle del PLC, i “L’Exit” que s’executarà l’últim cicle. També hi ha la finestra de “Variables.var” i “Global.typ”, que serà on declararem les nostres variables locals o globals i que més endavant les veurem. Les variables globals són les que faríem servir en comú si féssim diferents programes en el mateix projecte, i les variables locals les que faríem servir en un programa en concret del projecte. Es pot veure tot a la Figura 29.

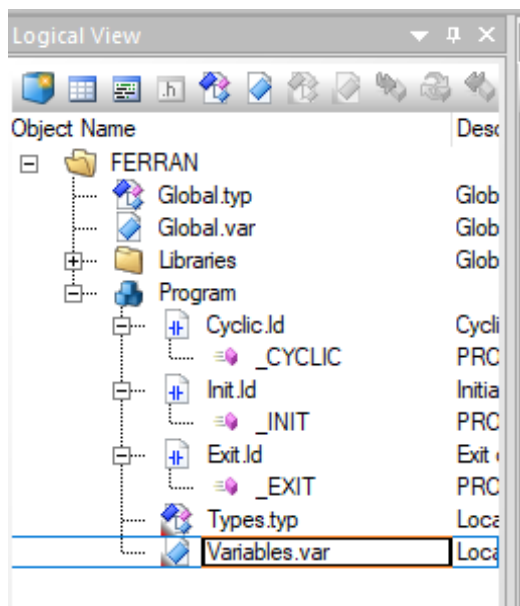


Figura 29. Imatge finestres del programa

Per poder veure les entrades/sortides físiques disponibles que tenim en els nostres mòduls, haurem d’entrar a la finestra de “Physical view”, com podem veure a la Figura 30.



Automàticament se'ns desplegaran les finestres de les connexions i dels mòduls que disposem. En aquest cas hem “clicat” al mòdul X3 i se'ns han desplegat totes les sortides que tenim en aquest mòdul.

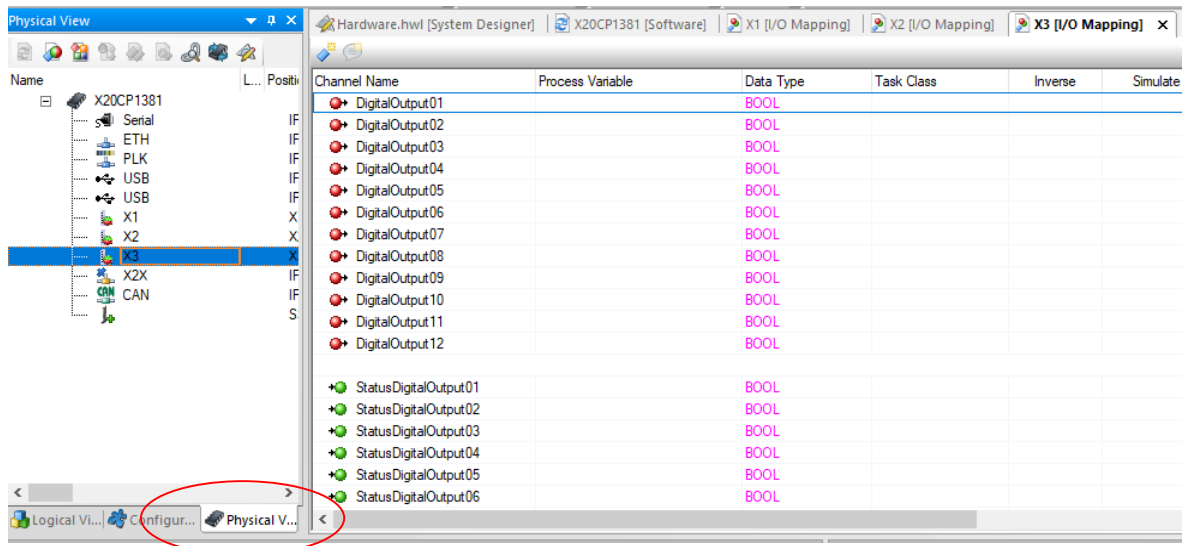


Figura 30. Physical view

En el cas que vulguem forçar alguna entrada/sortida amb alguna variable, que hauréem declarat anteriorment, tant global com local, hauréem de seleccionar-la fent-li “doble clic” a sobre a la finestra “process variable”, com es mostra a la Figura 31.

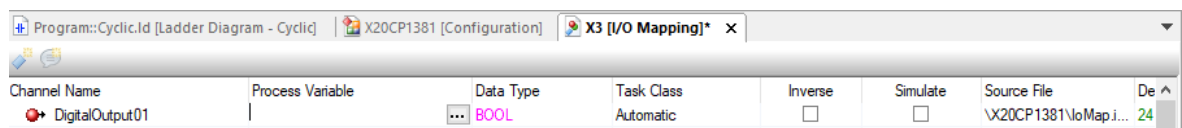


Figura 31. Per forçar variable amb la sortida que volem

Seguidament se'ns obrirà una finestra on podrem associar la variable que desitgem amb una sortida física. En aquest cas no hem definit cap variable global, per tant seleccionarem una variable local dins del “Program”, que serà en aquest cas una variable booleana que hem anomenat V1 i que correspondrà a una electrovàlvula. Els passos es poden observar a la Figura 32.

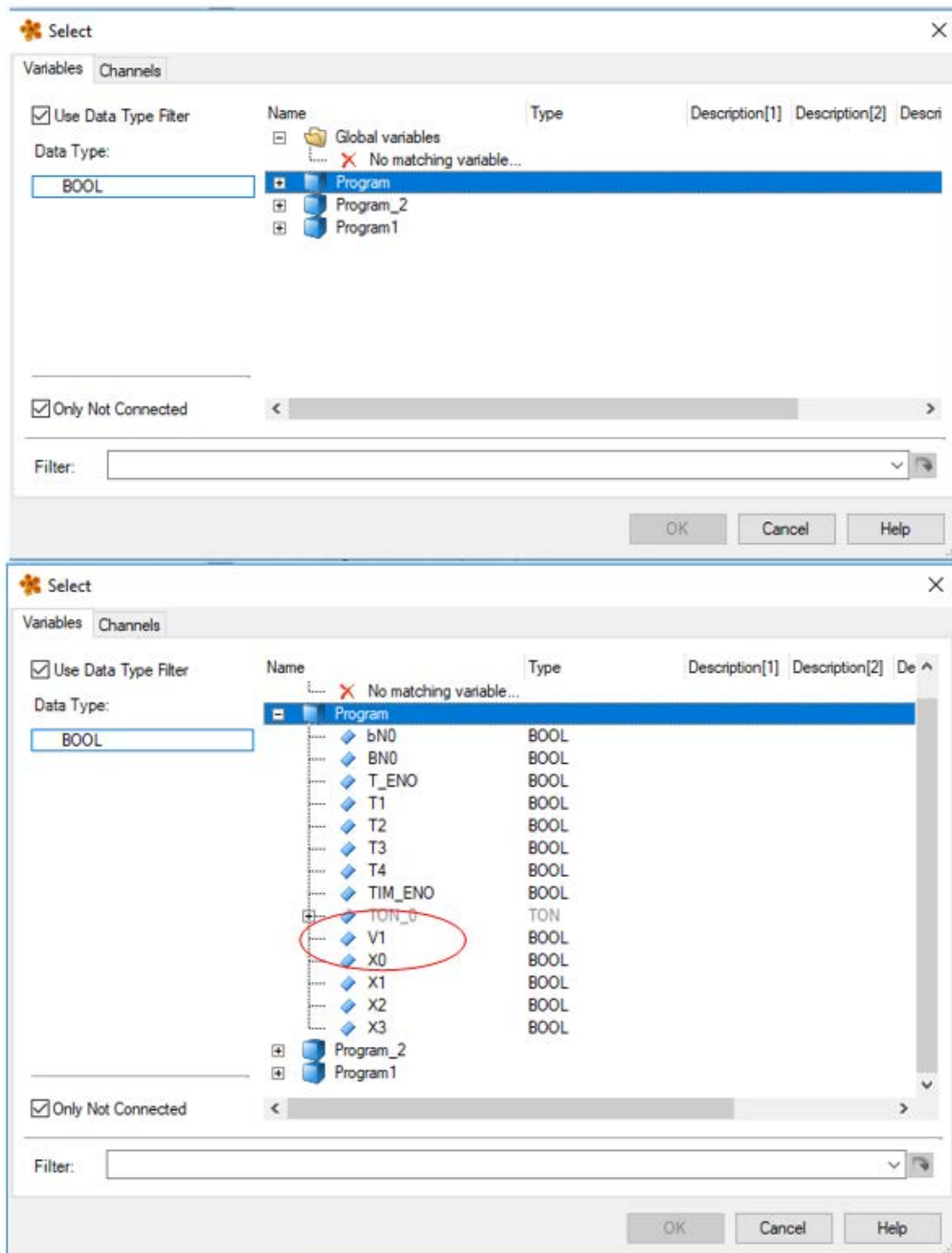


Figura 32. Selecció de la variable que volem mapejar amb la sortida física

I així amb totes les variables que vulguem mapejar amb qualsevol entrada/sortida física dels diferents mòduls de què disposem.

A l'hora de mapejar variables amb entrades/sortides físiques, haurem de tenir la precaució de què a les entrades/sortides digitals (DI/DO) hauran de ser mapejades amb variables de tipus booleà, tot i que ens deixaria amb un altre tipus de variable. I amb les entrades/sortides analògiques el mateix, el tipus de variable haurà de ser tipus INT (integer).

Un altre apartat molt important és la finestra de la CPU on hi ha tots els "Cycle time" (temps de cicle) de la CPU entre els quals podem triar per ubicar-hi els nostres programes. Ho podem veure a la Figura 33. Per defecte els posa al cycle time de 100 ms.

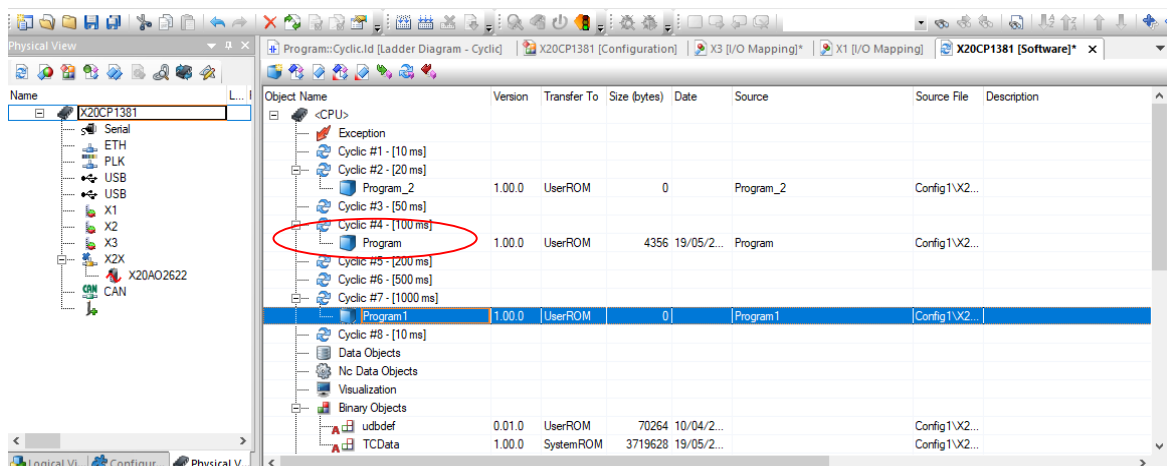


Figura 33. CPU

"Clicarem" sobre el programa que vulguem i l'arrossegarem cap al "cycle time" que desitgem. Si volem habilitar o deshabilitar els programes, per tal que a l'hora de carregar-lo al PLC només carregui el que desitgem, "clicuem" amb botó dret sobre el programa que vulguem habilitar/deshabilitar i marquem "enable" (habilitar) o disable (deshabilitar). Ho podem veure a la Figura 34.

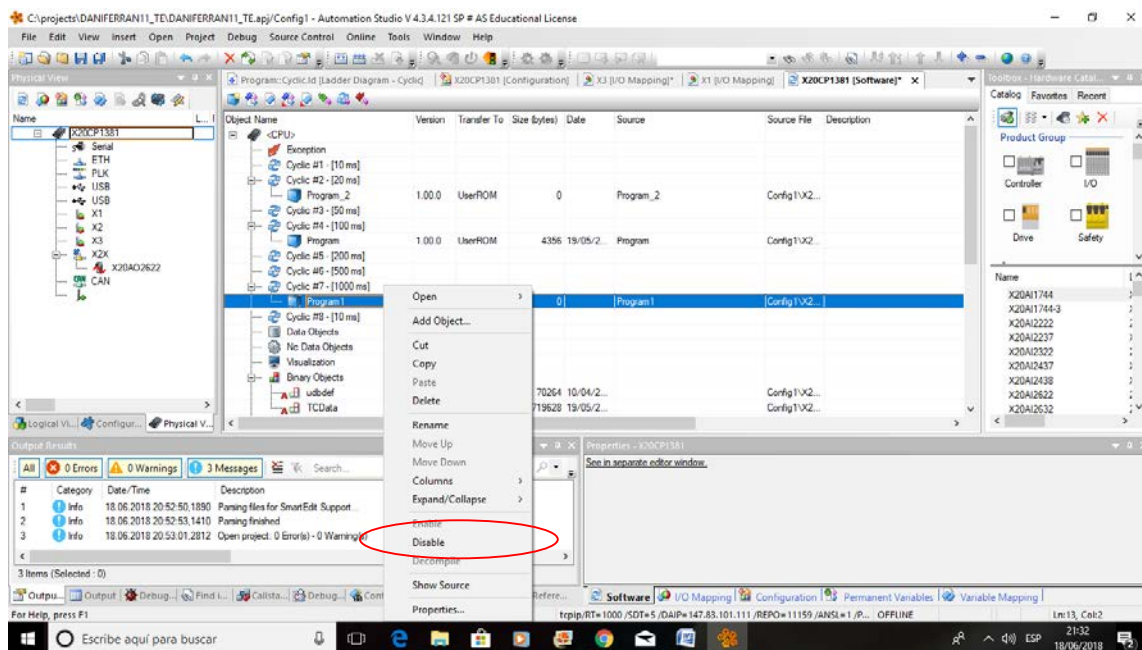


Figura 34. Enable/disable del programa

## 5.2. PROGRAMACIÓ

Un cop ens hem familiaritzat amb l'Automation Studio podem començar a programar el nostre programa. El primer que haurem de fer serà connectar-nos a través del nostre ordinador amb el PLC. Per saber la IP del PLC haurem d'anar a "Physical view", "cliquejar" amb el botó dret a la finestra "ETH" i anar a "configuration", com ens mostra la Figura 35:

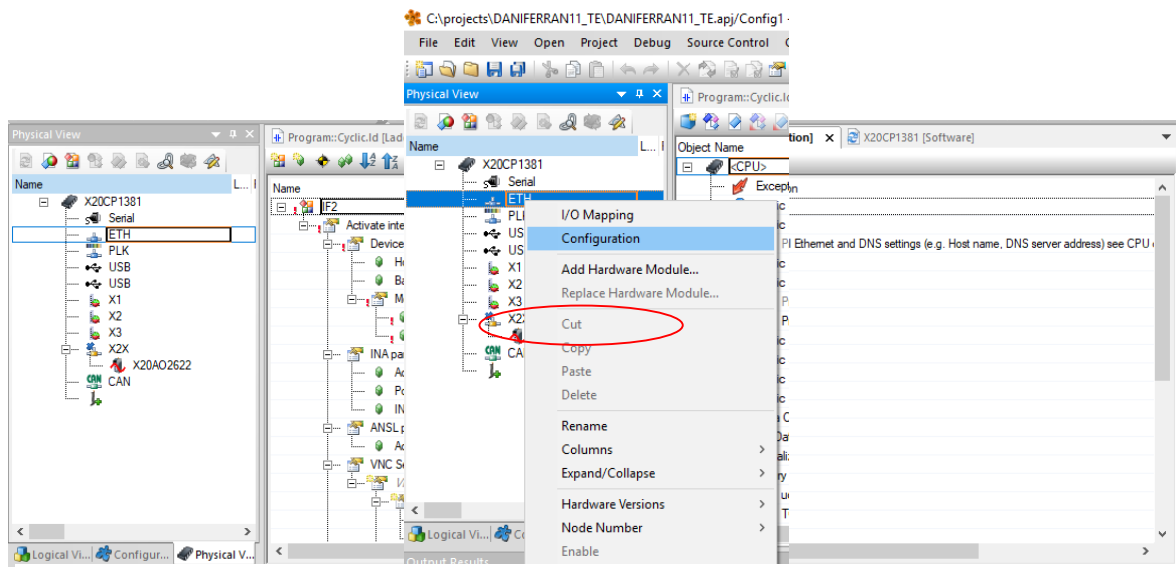


Figura 36. IP adress

Seguidament se'ns obrirà la finestra que es veu a la Figura 36 i a la "IP adress" veurem la IP del nostre PLC.

Lavors haurem de modificar la direcció IP configurada a la interfície d'Ethernet de l'ordinador (que ha de ser diferent del PLC però ha d'estar al mateix rang de subxarxa). Per exemple, la IP del nostre PLC és 147.83.101.111, i la màscara de subxarxa 255.255.255.0.

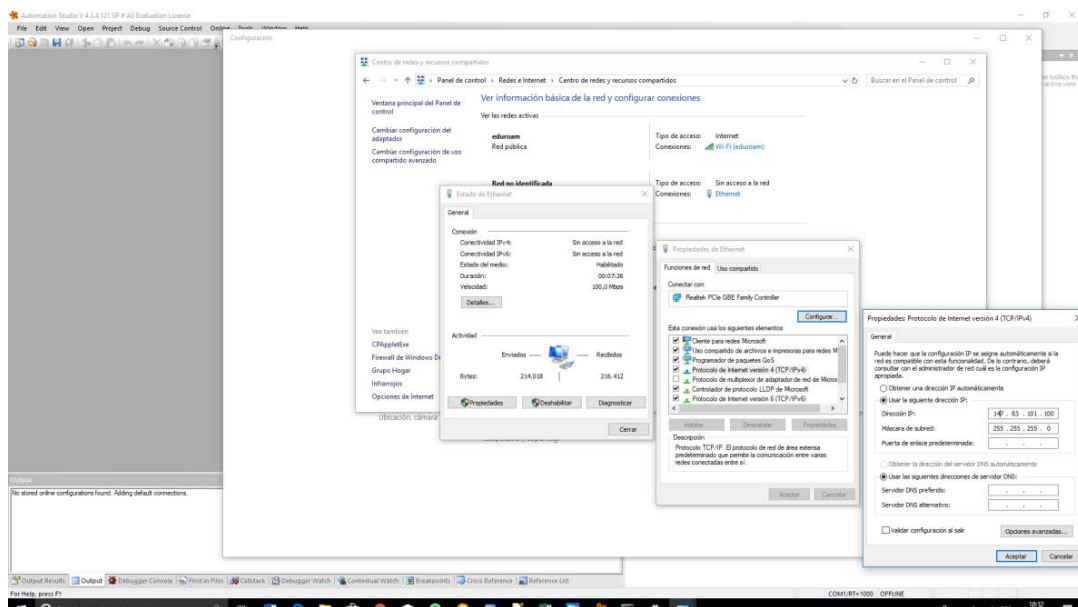


Figura 37. Canviar IP ordinador

L'ordinador ha d'estar a qualsevol direcció 147.83.101 "X", on "X" és qualsevol nombre sencer entre 1 i 254, menys la 111. Per configurar la IP del nostre ordinador haurem d'anar a panel de control, "redes" i internet, "centro de redes" i "recursos compartidos". Fer doble clic a ethernet, i anar a "propiedades". Fer doble clic a IPv4 i modificar la IP i la màscara de subxarxa, com es pot veure a la Figura 37.

Per assegurar que hem configurat bé la IP, realitzarem la connexió entre l'ordinador i el PLC. Obrirem la finestra de la part superior "Online" i obrirem la finestra "Settings", com ens mostra la Figura 38.

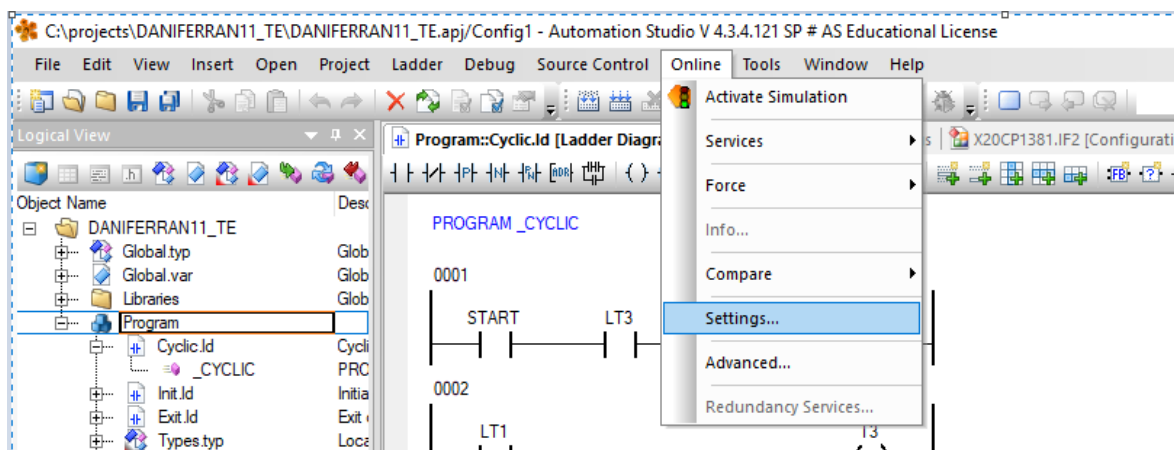


Figura 38. Online -> Settings

Se'ns obrirà una finestra on sortirà la nostra TCP/IP amb la IP del nostre PLC. En primer lloc haurem de desplegar la finestra "browse" perquè ens trobi la nostra CPU, com veiem a la Figura 39.

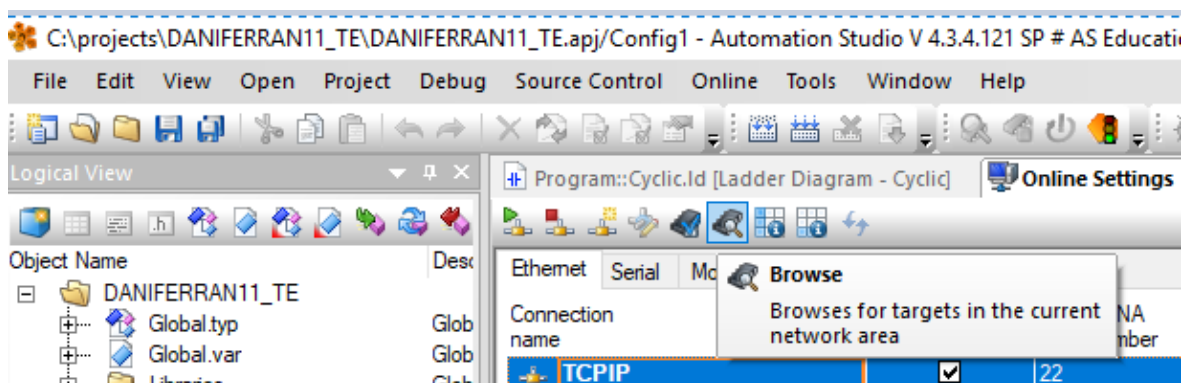


Figura 39. Browse

Quan ens hagi trobat la CPU clicarem amb el botó dret a "TCPIP i seguidament a "connect",  
Figura 40.

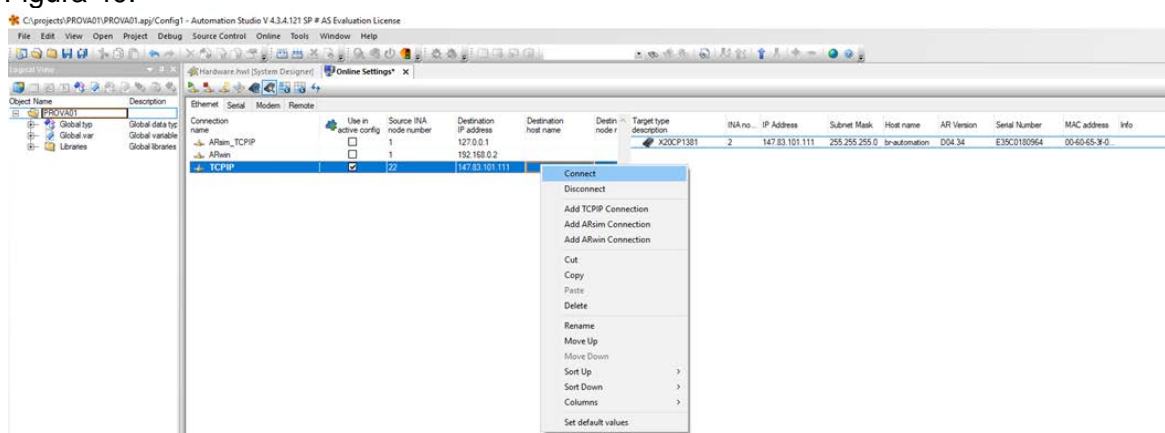


Figura 40. Connect

Sabrem que ens hem connectat correctament perquè l'estat del PLC haurà canviat de "OFFLINE" a "RUN" a la part inferior del programa (vegeu Figura 41). Ara ja estem connectats.

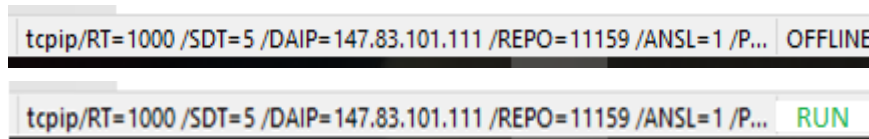


Figura 41. OFFLINE-> RUN

### 5.3. Programació del PLC en llenguatge Ladder

A continuació utilitzarem com a exemple de programació el Graficet de la Figura 66.

El llenguatge de programació utilitzat és "Ladder". Haurem d'incorporar al nostre projecte un programa tipus LD com hem descrit a la Figura 28. Per programar el Graficet seguirem les següents etapes:

- Activarem les transicions, anomenades  $T_i$  amb  $i = 1, 2, \dots, m$ , segons les condicions imposades per passar d'una etapa a una altra etapa.
- Un cop activades les transicions, cal desactivar (RESET) l'etapa anterior i activar l'etapa següent (SET). Les etapes s'anomenen  $X_i$ , amb  $i = 1, 2, \dots, n$ .

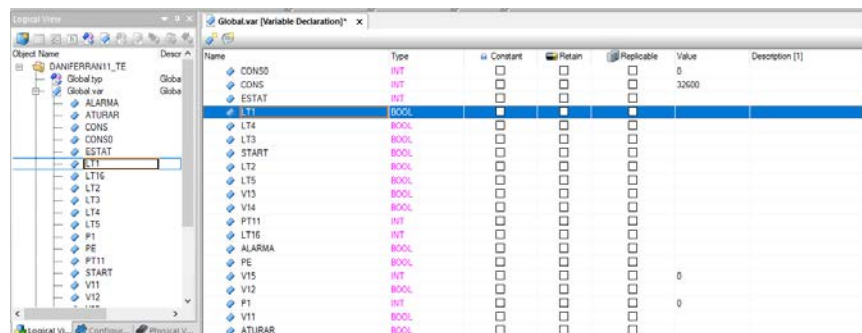


- Les etapes activaran els actuadors que desitgem en aquella etapa.

### 5.3.1. Definició de variables

El primer que haurem de fer és definir les variables globals i locals. Com a variables globals definirem les variables comunes les utilitzades a tots els programes i les locals les pròpies de cada programa.

Dins de la finestra de variables globals (tal com hem descrit a la Figura 29), definirem els sensors, actuadors i les constants que utilitzarem per donar-li el valor als actuadors analògics (P1 i V15). Com que el programa no deixa anomenar elements amb punts, la nomenclatura que utilitzarem serà la següent: els sensors digitals de nivell en lloc d'anomenar-los LT 1.1, LT 1.2, etc... els anomenarem LT1, LT2, etc... Les electrovàlvules en lloc de V 1.1, V1.2, etc... les anomenarem V11, V12, etc... La motobomba en lloc de P1.1 l'anomenarem P1. El sensor de pressió PT 1.1 l'anomenarem PT 1. Tots els sensors, polsadors, LEDs i les electrovàlvules digitals (tot o res), les definirem com variables de tipus booleà. I la motobomba, l'electrovàlvula V15, les constants i el sensor, que son tipus analògiques, les definirem com a variables de tipus Integer (de 16 bits). Ho podem veure tot a la Figura 42.



Name	Type	is Constant	Retain	Replicable	Value	Description [1]
CONS0	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
CONS	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32600	
ESTAT	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT1	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT4	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT3	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
START	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT2	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT5	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
V13	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
V14	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
PT11	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
LT16	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ALARMA	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
PE	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
V15	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
V12	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
P1	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
V11	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ATURAR	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Figura 42. Definició variables globals



Com a variables locals hi haurà les transicions i etapes dels Graficets, els temporitzadors i les variables utilitzades per memoritzar la lectura d'algun dels detectors analògics, com seria el cas del valor del sensor LT16 que s'utilitza per realitzar comparacions. Les etapes i les transicions són booleans, el temporitzador tipus TON. Les Figura 44, Figura 45, Figura 47 i Figura 48 mostren com han quedat mapejades les entrades i sortides físiques en els diferents mòduls.

Un cop tenim les variables definides farem el mapeig de totes les entrades/sortides de les quals disposem, com hem descrit a la Figura 30 i a la Figura 31.

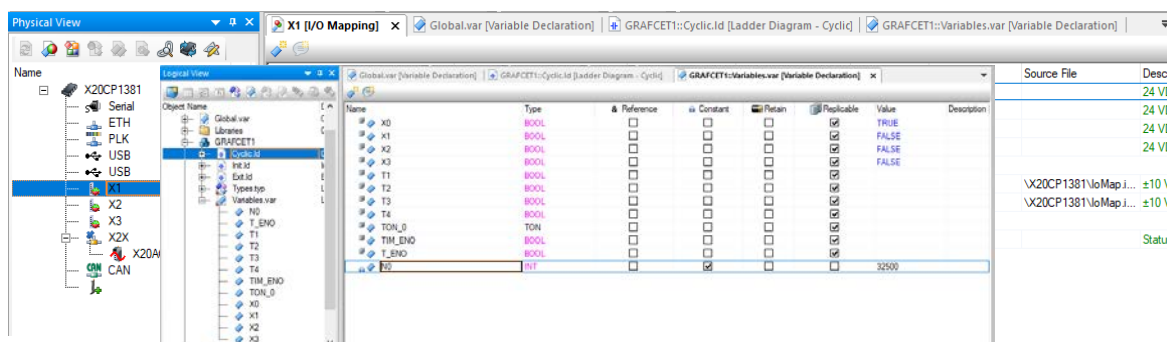


Figura 43. Definició variables locals

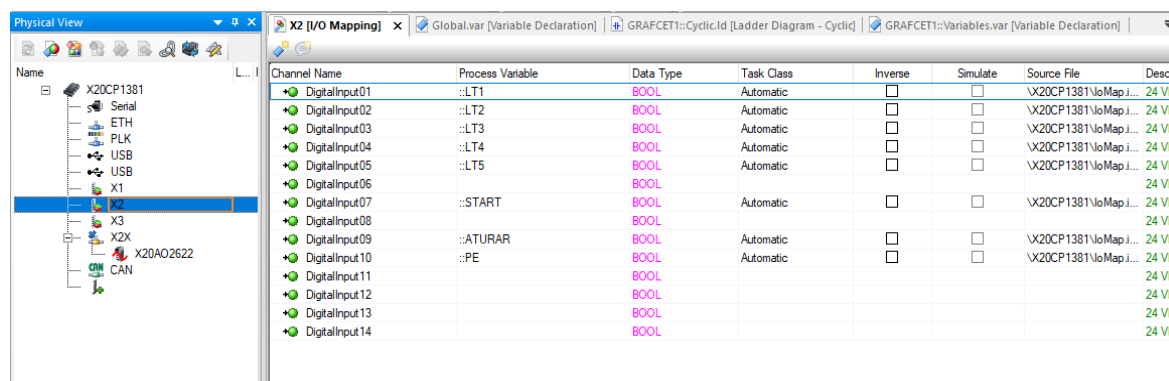
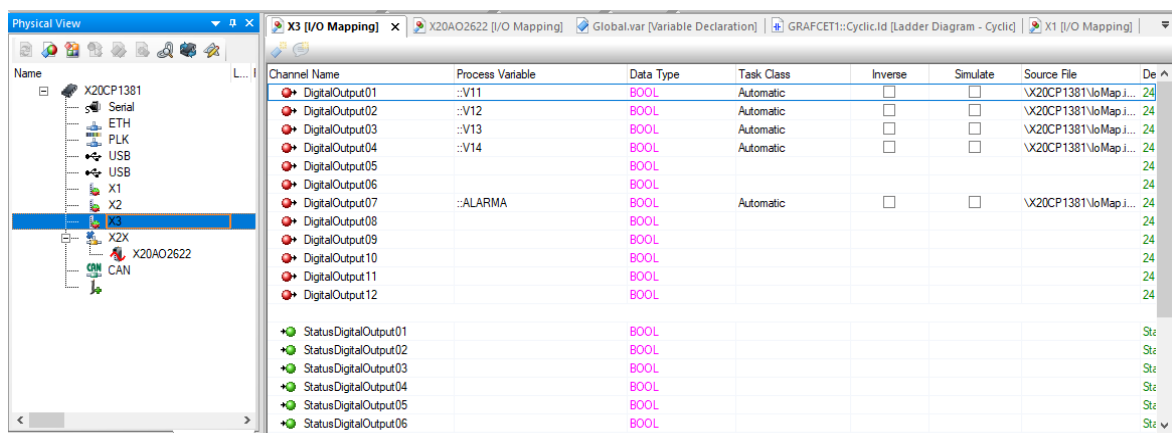
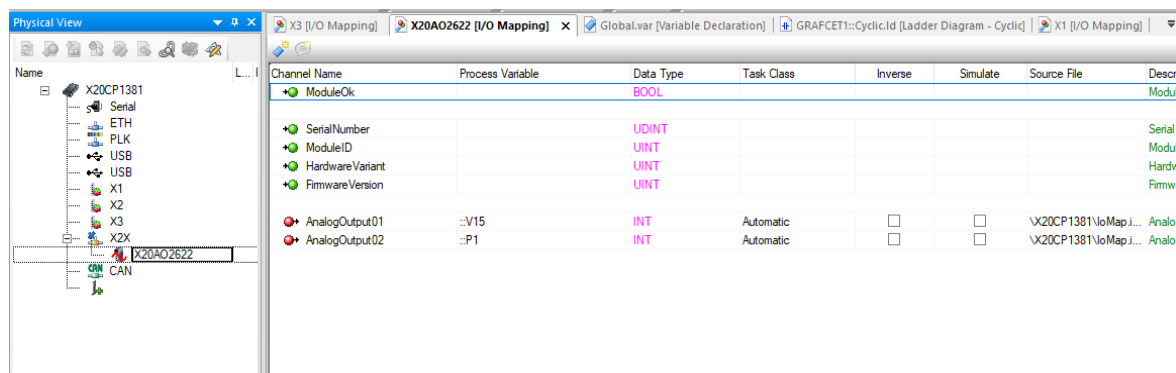


Figura 45. Mapeig entrades digitals del mòdul X2



Channel Name	Process Variable	Data Type	Task Class	Inverse	Simulate	Source File	De
DigitalOutput01	::V11	BOOL	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	24
DigitalOutput02	::V12	BOOL	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	24
DigitalOutput03	::V13	BOOL	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	24
DigitalOutput04	::V14	BOOL	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	24
DigitalOutput05		BOOL					24
DigitalOutput06		BOOL					24
DigitalOutput07	::ALARMA	BOOL	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	24
DigitalOutput08		BOOL					24
DigitalOutput09		BOOL					24
DigitalOutput10		BOOL					24
DigitalOutput11		BOOL					24
DigitalOutput12		BOOL					24
StatusDigitalOutput01		BOOL					Stz
StatusDigitalOutput02		BOOL					Stz
StatusDigitalOutput03		BOOL					Stz
StatusDigitalOutput04		BOOL					Stz
StatusDigitalOutput05		BOOL					Stz
StatusDigitalOutput06		BOOL					Stz

Figura 46. Mapeig sortides digitals del mòdul X3



Channel Name	Process Variable	Data Type	Task Class	Inverse	Simulate	Source File	Descri
ModuleOk		BOOL					Modul
SerialNumber		UDINT					Serial i
ModuleID		UINT					Modul
HardwareVariant		UINT					Hardw
FirmwareVersion		UINT					Fimwv
AnalogOutput01	::V15	INT	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	Analog
AnalogOutput02	::P1	INT	Automatic	<input type="checkbox"/>	<input type="checkbox"/>	\X20CP1381\IoMap.i...	Analog

Figura 47. Maig actuadors analògics del mòdul X4

### 5.3.2. Programació en Ladder

Per programar amb ladder, al ser un diagrama de contactes o relés, haurem de saber que hi ha símbols similars als elèctrics, com per exemple el contacte normalment obert/tancat. A la Figura 48 podem veure un exemple de contacte obert. Quan la variable associada a un contacte obert val 1 el contacte es tanca, i quan la variable associada a un contacte normalment tancat val 1, el contacte s'obre.

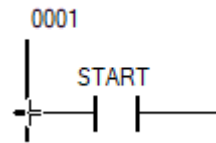


Figura 48. Contacte obert

Per a qualsevol element que li vulguem donar un nom, hem de fer un doble clic a sobre de l'element i ja podrem anomenar-lo.

Per activar variable lògica utilitzarem una bobina. Quan els contactes que l'han d'activar estan actius, aquesta bobina estarà a 1 (true). Quan deixin d'estar a 1, la bobina deixarà d'estar activa. Com es pot veure a la Figura 49, START i LT3 estan en sèrie, per tant quan el botó START = 1 (està actiu) i el nivell LT3 detecta senyal, la transició T1 estarà a 1, i quan START o LT3 deixin d'estar a 1, T1 passarà a 0.

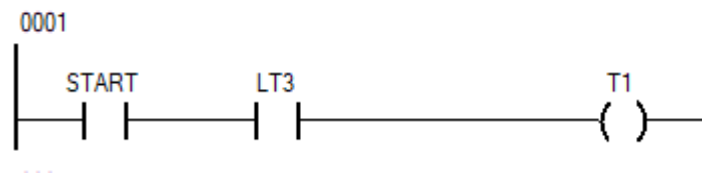


Figura 50. Activació bobina T1

Una altra manera d'activar variables, és amb el SET, i amb el RESET per desactivar-les. Quan li donem un pols a SET la variable que hi ha associada passarà a valdre 1, fins que li fem un pols al RESET. A la Figura 50 podem veure que quan l'etapa X0 = 1 i T1 = 1, farem un SET a l'etapa X1 i un RESET a X0.

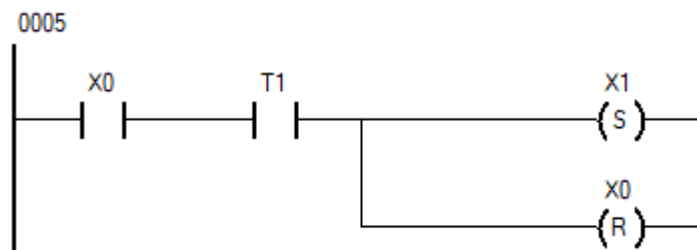


Figura 49. SET i RESET

Per als actuadors analògics no es poden utilitzar les bobines, ja que necessiten un valor de 0 a 32767 al ser variables tipus integer de 16 bits. Per tant, utilitzarem el FB MOVE, és un

bloc funcional que permet assignar un valor a una variable o d'una variable a un altre. Per utilitzar el MOVE haurem d'anar a la finestra "FB" (function blocks), buscar-lo i seleccionar-lo, (com mostra la Figura 51), i ja ens sortirà al nostre programa.

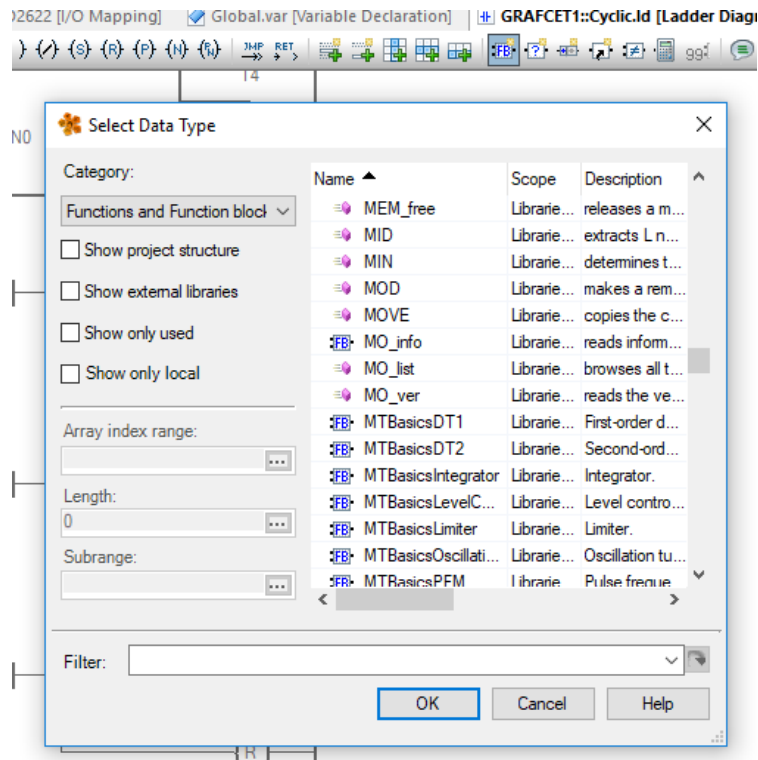


Figura 51. Selecció MOVE

El MOVE té dues entrades, "EN" que és on haurem d'introduir la condició perquè es faci el MOVE, i l'entrada on introduïrem el valor que volem escriure. I les dues sortides de què disposa són "ENO", i on volem moure el valor que hem introduït. A la Figura 52 podem veure un FB MOVE: en el qual la condició perquè es realitzi l'operació és que l'etapa X2 estigui activa. Llavors passarem el valor CONS (que l'hem definit prèviament a les variables globals) cap a la motobomba P1. Per anomenar les entrades/sortides del MOVE li hem de fer un doble clic i posar el nom que desitgem. A la Figura 52 podem veure un exemple d'un MOVE.

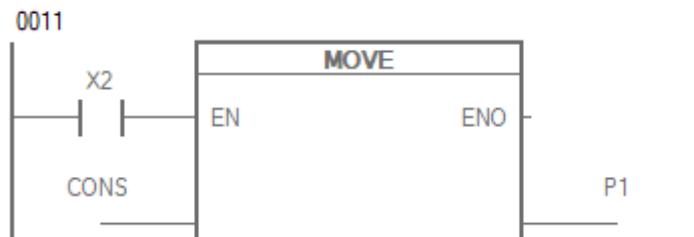


Figura 52. MOVE

Un altre element que farem servir en aquest programa són les comparacions de les variables, <, >, >=, <=, <> o =. Aquestes les realitzarem amb la funció COMPARE. A l'entrada del COMPARE hi posarem un contacte normalment obert anomenat TRUE que ja està definit al mateix sistema i que sempre serà 1, per tant a cada cicle del programa farà la comparació. A l'altra entrada posem la comparació a realitzar. Quan aquesta comparació és certa, s'activarà la sortida T4 del comparador. A la Figura 53 podem veure una comparació a on LT16 ha de ser major que N0, mentre aquesta comparació sigui certa, T4 estarà activa.

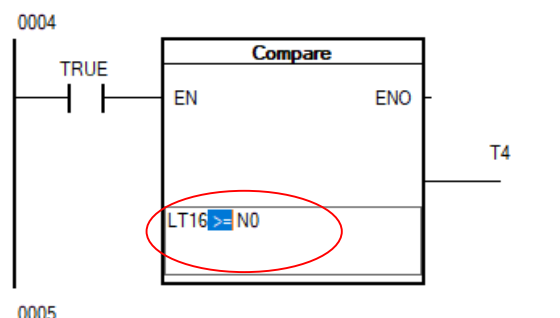
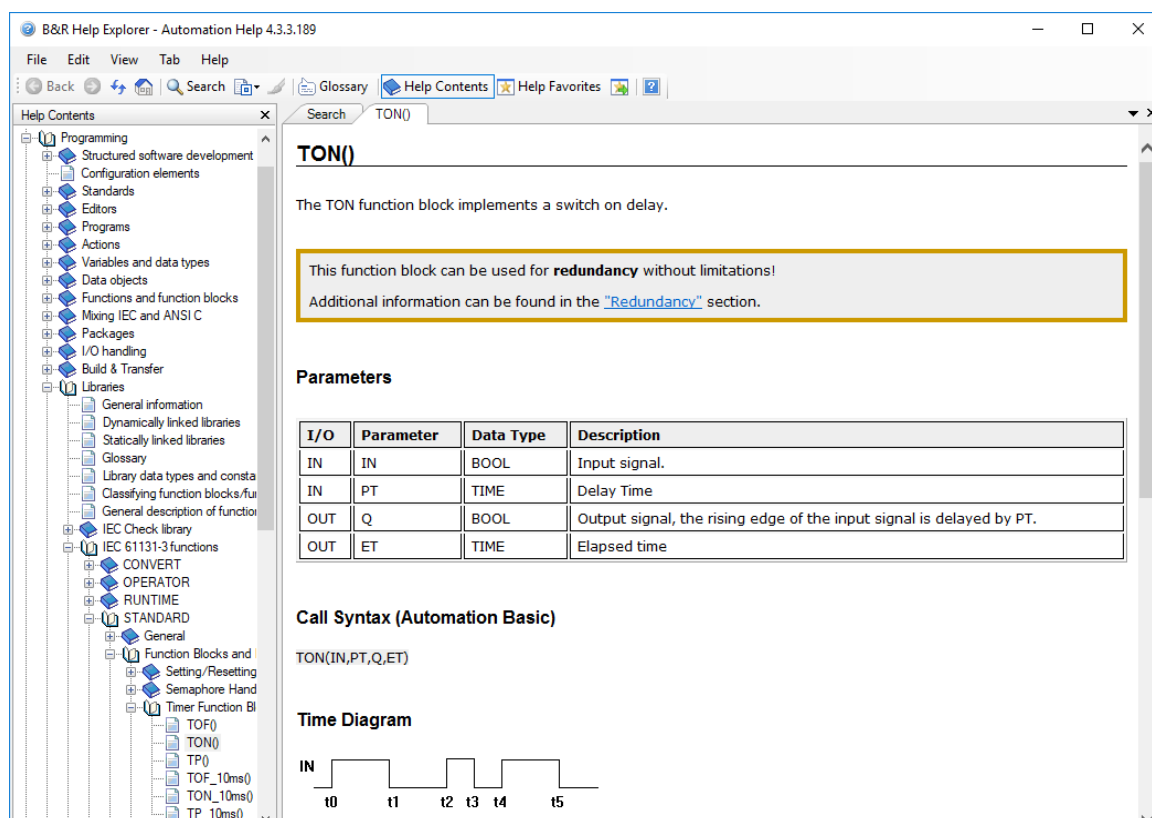


Figura 53. COMPARE

Un altre element que hem de conèixer és el temporitzador TON, que el trobarem a les llibreries de "FB" com el COMPARE i el MOVE. Són temporitzadors que quan s'activa comença el comptatge i al finalitzar activa la seva variable de sortida. Hem de saber que l'entrada de temps treballa en base de 10 ms. Per habilitar el temporitzador farem servir el contacte TRUE, a l'entrada PT li entrarem el valor de temps que volem en ms. A l'entrada IN li entrarem la condició amb la qual volem que el temporitzador comenci a comptar. I a la sortida Q la variable que volem activar quan el temporitzador acabi de comptar. A la Figura 54 veiem el temporitzador que comença a actuar quan X1 és 1, i quan acaba de comptar activa T2.

Si cliquem a F1 tenint seleccionat qualsevol element o funció, automàticament se'ns obrirà l'ajuda de l'element o funció, on ens explicarà el seu funcionament i com definir-lo. La Figura 55 mostra l'ajuda del temporitzador.



**TON()**

The TON function block implements a switch on delay.

This function block can be used for **redundancy** without limitations!  
Additional information can be found in the "[Redundancy](#)" section.

**Parameters**

I/O	Parameter	Data Type	Description
IN	IN	BOOL	Input signal.
IN	PT	TIME	Delay Time
OUT	Q	BOOL	Output signal, the rising edge of the input signal is delayed by PT.
OUT	ET	TIME	Elapsed time

**Call Syntax (Automation Basic)**

TON(IN,PT,Q,ET)

**Time Diagram**

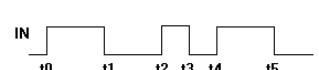


Figura 55. Ajuda del temporitzador



## 5.4. SIMULACIÓ

Un cop tenim el programa fet (a l'Annex 1, trobarem tots els programes), l'haurem de compilar per veure si hi ha algun error. Per compilar haurem de seleccionar la finestra Build o prémer F7, com podem veure a la Figura 56.

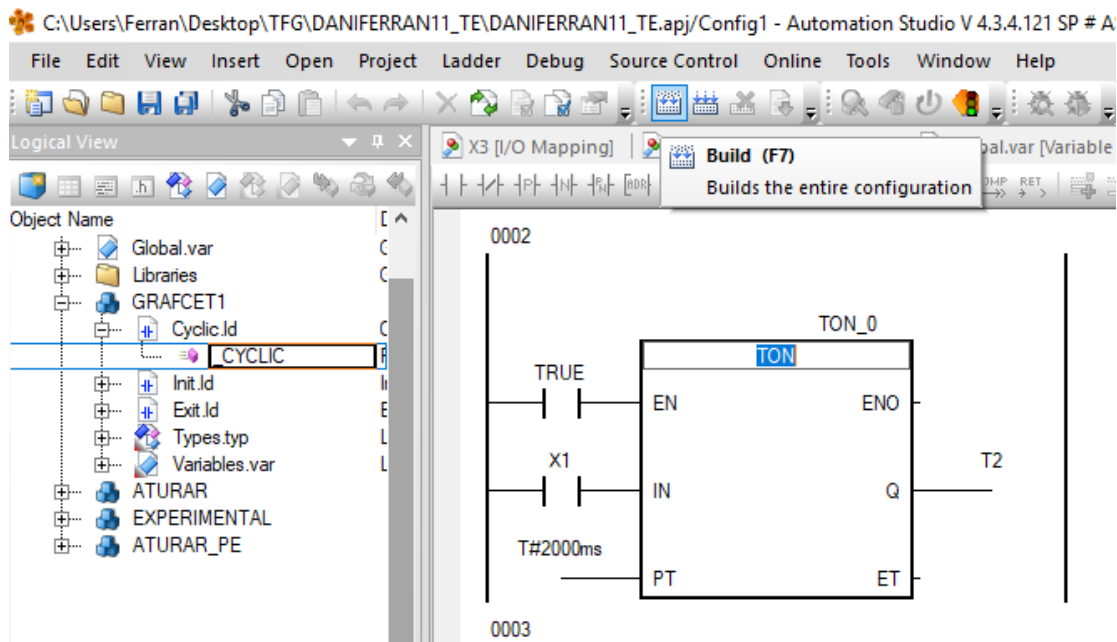


Figura 56. Compilar

Quan acabi de compilar, ens dirà si hi ha hagut algun error. Podem veure a la Figura 57 la finestra del programa compilat amb un error. Si l'obrim, ens diu on són i podrem resoldre el problema.

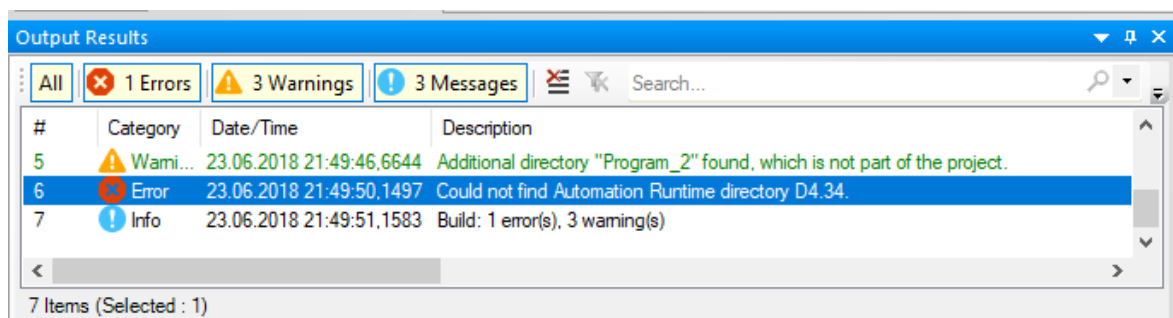


Figura 57. Programa compilat amb error

Un cop hem compilat el programa, i no ens ha donat cap error, com es veu a la Figura 58, podrem simular-lo.



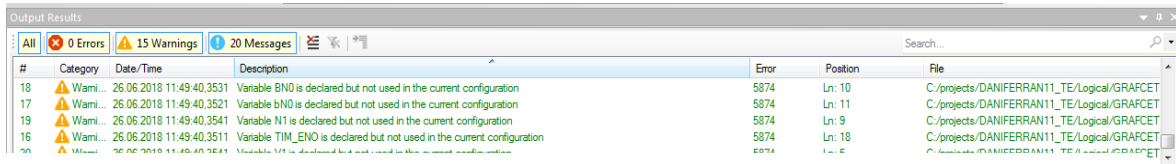


Figura 58. Compilat OK

Per començar a simular haurem d'activar el mode de simulació, clicant al semàfor de la finestra com es veu a la Figura 59. En el mode simulació ens connectem amb una màquina virtual simulant la connexió amb el PLC.

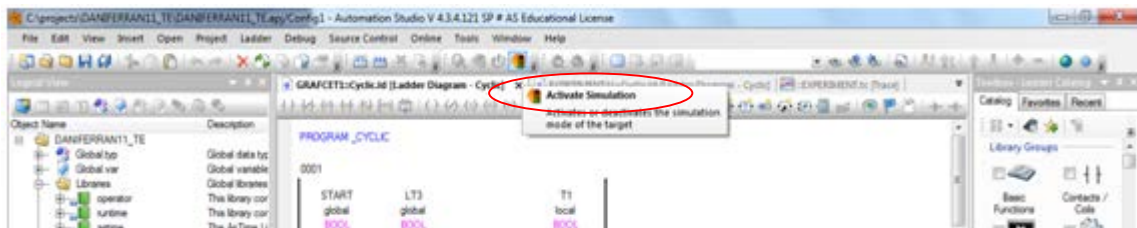


Figura 59. Activar mode simulació

Per comprovar que ens connectem correctament l'estat de la màquina virtual haurà de quedar en estat RUN com es mostra a la Figura 60.

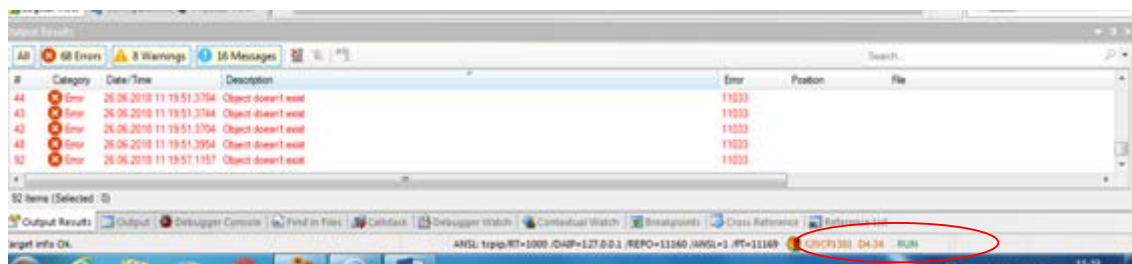


Figura 60. En mode RUN

Un cop la màquina virtual està en estat RUN podem transferir el programa clicant a la icona transfer com ens mostra la Figura 61.

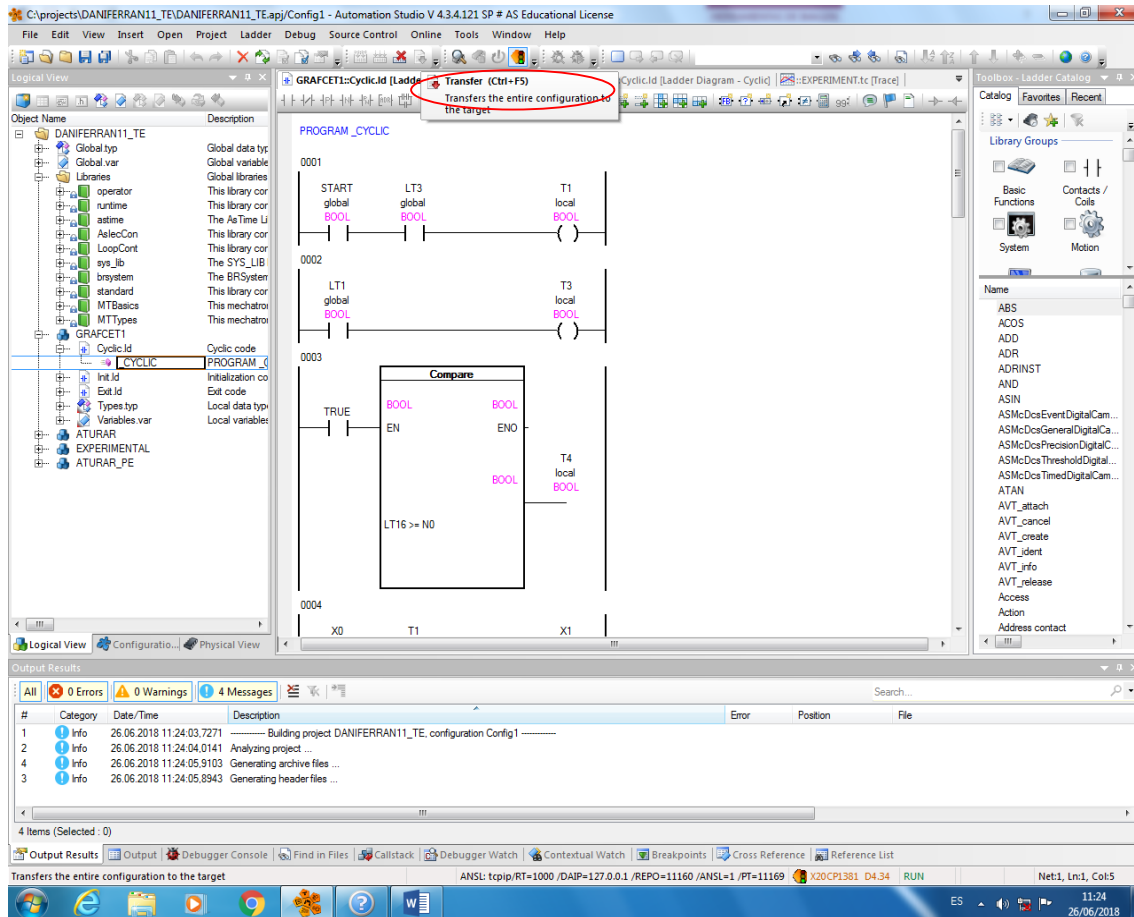


Figura 61. Transfer

Automàticament ens preguntarà si volem transferir el programa. Clicarem a transferir com mostra la Figura 62.

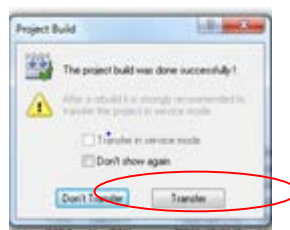


Figura 62. Transferir programa



Llavors ens transferirà el programa a la màquina virtual. Quan finalitzi la transferència, com ens mostra la Figura 63, ja podrem començar la simulació.

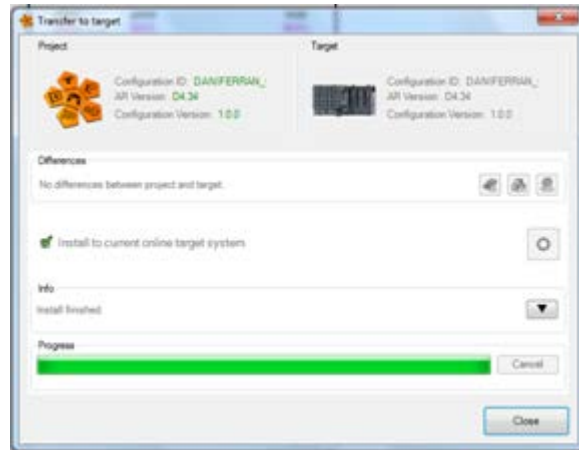


Figura 63. Transferència realitzada

Per poder veure l'estat actual de les variables clicarem a la finestra watch (lupa), com es mostra la a Figura 64.

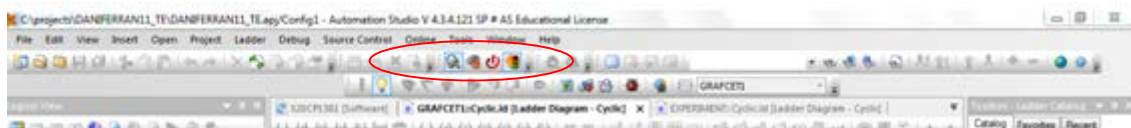


Figura 64. Watch

Se'ns obrirà la pantalla del programa online on podrem incloure les variables que vulguem, per poder forçar-les i simular la seqüència que vulguem. Per incloure les variables farem botó dret, i "add" variables com mostra la Figura 65. I amb el botó de CTRL de l'ordinador i seleccionant totes les variables que vulguem i clicant "add", ens sortiran les variables en l'estat actual, com mostra la Figura 66.

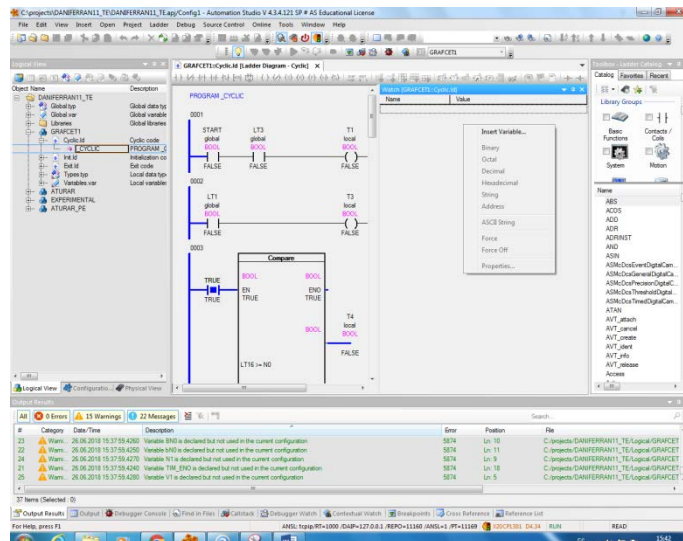


Figura 65. Add variables

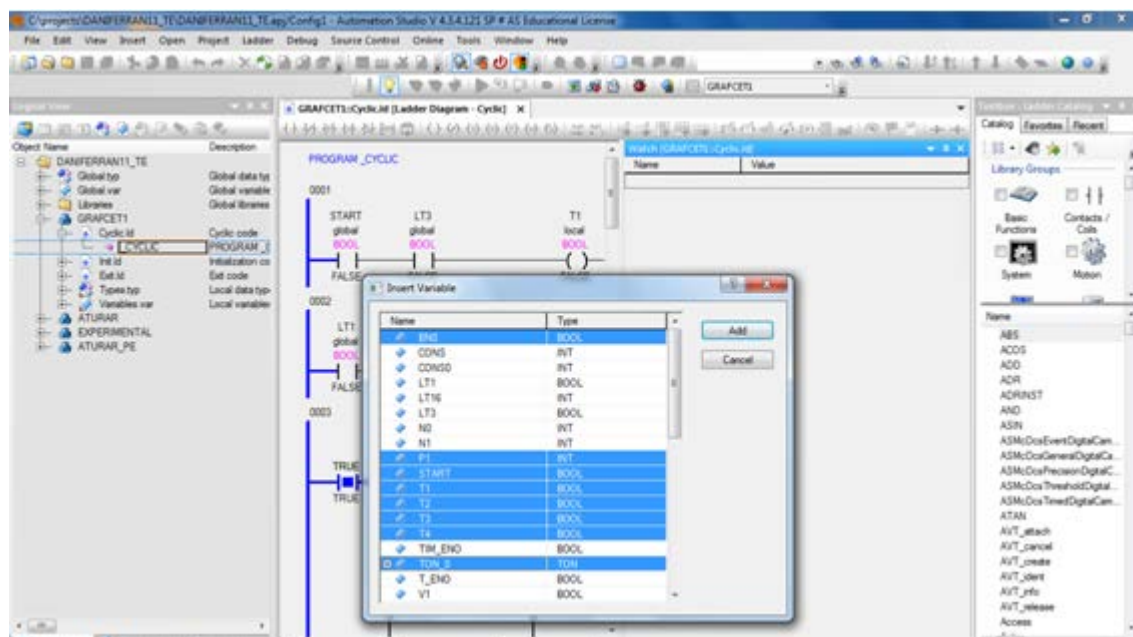


Figura 66. Selecció de variables

Un cop podem veure l'estat de les variables online, ja podem forçar el valor i realitzar la simulació del programa fent la seqüència que desitgem. Per forçar el valor de qualsevol variable clicarem a sobre del seu value i posarem el valor que vulguem. Com a exemple,

la Figura 67, mostra com hem canviat el valor de START i LT3 per començar la seqüència. Cal dir que el rang de variació de les variables analògiques és de 0 a 32667. A la figura també es mostra l'estat de totes les variables.

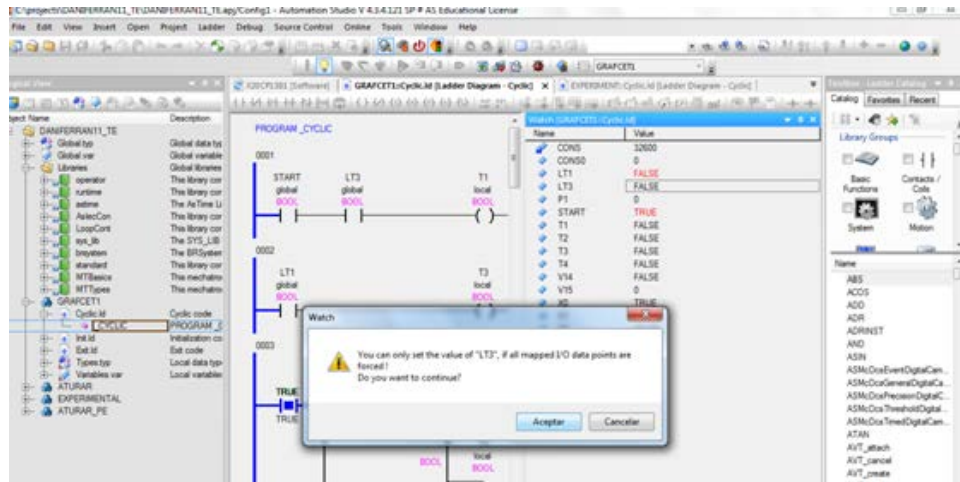


Figura 67. Forçar variables

## 5.5. Connectats amb el PLC

Per treballar connectats amb el PLC, el primer que s'ha de fer és sortir del mode simulació. Després connectarem el cable ethernet i comprovarem que ens connecta PLC i ordinador compilant i transferint el programa de la mateixa manera que al mode simulació. Un cop connectats amb el PLC ja podrem provar el programa i les seves seqüències amb les entrades i sortides físiques, comprovant amb la finestra Watch que tot funciona correctament com indica el graficet. En tot moment podem veure en quina etapa ens trobem, les transicions i les condicions d'entrada per passar d'una etapa a un altre observant a la finestra del Watch l'estat de totes les variables declarades al programa.

## 6. AUTOMATITZACIÓ DE L'ESTACIÓ

Per a l'aprenentatge de l'entorn de programació i del llenguatge del PLC es proposa la realització de diferents activitats pràctiques. Aquestes activitats proposades tenen un grau creixent de dificultat i s'han agrupat en dos blocs, el primer consisteix a tractar el procés com un sistema d'esdeveniments discrets i el segon inclou el tema de regulació de variables.

### 6.1 Sistema d'esdeveniments discrets

Ens plantegem automatitzar el procés dissenyat considerant diferents graus de complexitat. Concretament, es vol automatitzar el procés considerant una seqüència bàsica, una seqüència controlada i, per últim, s'incorporaran mesures de seguretat.

#### 6.1.1. Seqüència bàsica

##### a) Objectius

- Programar una seqüència bàsica amb LD Program
- Aprendre el funcionament dels temporitzadors
- Llegir el detector analògic

##### b) Especificacions de disseny

- Cal omplir el tanc ALT fins a assolir el nivell indicat pel detector LT1
- Una vegada omplert, cal buidar-lo
- Com a condicions inicials cal que al dipòsit BAIX hi ha d'haver aigua suficient, és a dir, LT6 ha d'estar actiu, i el cicle s'inicia al primer START

##### c) Programació

La programació que permet garantir la seqüència de disseny s'observa en el Grafcet de la Figura 68, on la condició inicial serà que el tanc de baix tingui el nivell mínim, o sigui, que LT3 ha d'estar a 1, altrament no podrem iniciar la seqüència. Una altra condició per inicialitzar la seqüència és que s'ha de prémer el botó START. A part, el primer que farem és llegir el nivell inicial del tanc de dalt fent servir el sensor de nivell analògic LT 1.6, aquest valor es memoritza en la variable N0.

En prémer START, s'obrirà l'electrovàlvula V 1.4 i iniciarem el comptatge del temporitzador de 2 segons (aquest temporitzador es posa per estar segurs que la vàlvula de pas està

oberta abans d'activar la bomba). Quan el temporitzador hagi acabat de comptar passarem a la següent etapa on s'activarà la motobomba P1.1, i seguirem tenint oberta l'electrovàlvula V 1.4 enviant així l'aigua del tanc de baix al tanc de dalt. Anirem traspassant l'aigua d'un tanc a l'altre fins que arribi al nivell mínim del tanc de dalt (LT 1.1 actiu), i llavors un cop es compleixi aquesta condició pararem la motobomba i tancarem l'electrovàlvula V1.4, i obrirem l'electrovàlvula analògica V1.5, tornant així l'aigua del tanc de dalt al de baix fins que el nivell del sensor analògic LT 1.6 sigui més gran que N0.

El programa es mostra al Annex 1 i s'anomena Grafcet 1.

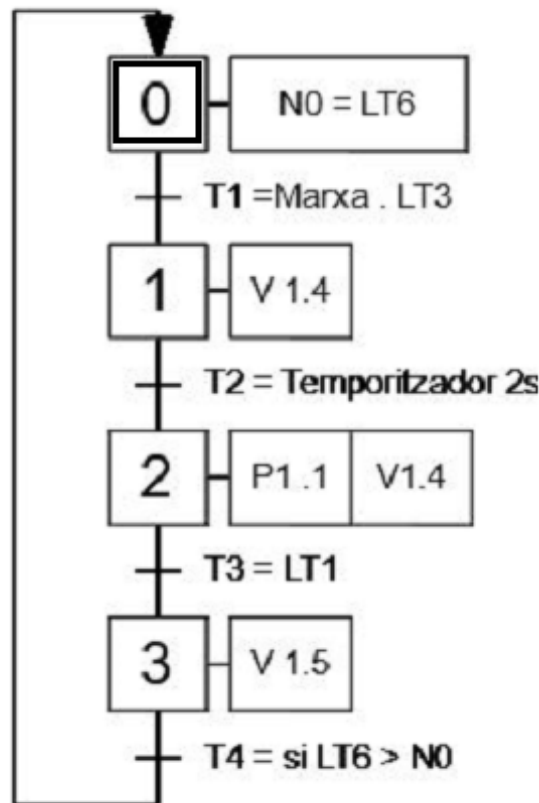


Figura 68. Grafcet seqüència bàsica

### 6.1.2. Seqüència indefinida

Pel que fa al segon nivell realitzarem el mateix procés que en el Grafcet anterior, amb la diferència que incorporarem un Grafcet d'aturada del sistema, i farem que la seqüència es



repeteixi de manera indefinida després de prémer el polsat el botó de marxa i es compleixin les condicions inicials. Durant tota la seqüència, si polsem el botó “Aturar” el procés seguirà fins arribar a l’etapa X0, l’ etapa inicial. Si volem tornar a començar la seqüència caldrà

pulsar un altre cop marxa i tornarà el procés automàtic. El Grafcet el podem veure a la Figura 69.

El programa es trobarà al Annex 1 i s’anomena ATURADA.

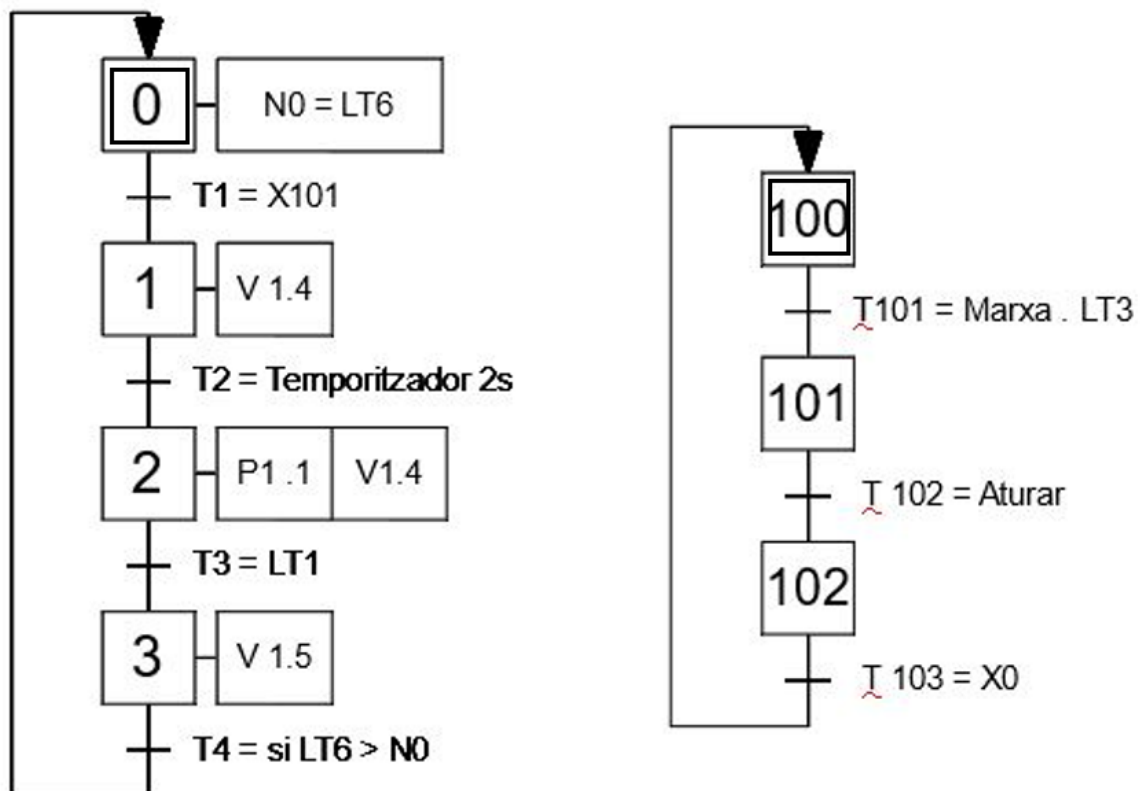


Figura 69. Grafcet Seqüència indefinida amb aturada

### 6.1.3. Seguretat

En el tercer nivell, serà el mateix que el d'aturada incorporant l'Aturada d'emergència. Si polsem l'interruptor de PE, parará tot el procés en sec, parant tots els actuadors a l'instant, i encenent el LED d'alarma indicant així que en el procés ha passat alguna cosa i s'ha hagut de prémer PE. Un cop traiem el PE, seguirà la seqüència des d'on ha quedat aturada fins a arribar de nou a l'etapa inicial, igual que en el programa d'aturada. L'alarma s'apagarà en el moment en què traiem el PE.





El programa de la Figura 70 es trobarà al Annex 1 i s'anomena ATURADA\_PE.

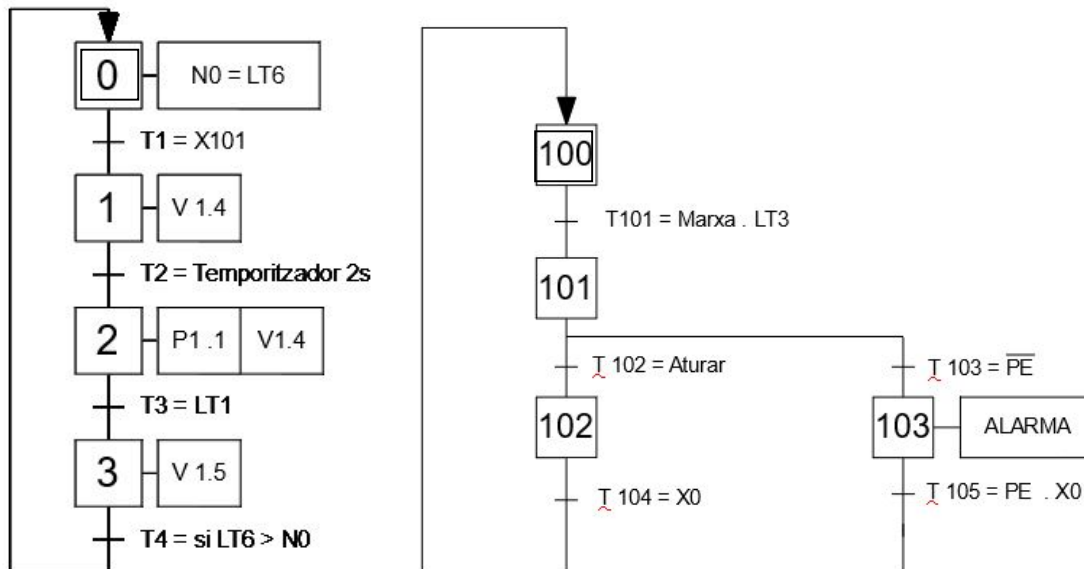


Figura 70. Grafcet automàtic amb aturada i PE.

Un punt important a tenir en compte a l'hora de programar serà quan vulguem realitzar comparacions amb variables analògiques, ja que per les oscil·lacions dels valors llegits dels sensors és molt difícil que la instrucció d'estrictament igual ( $=$ ) es compleixi, per tant intentarem programar amb  $\leq$  o  $\geq$  depenent de la finalitat que vulguem obtenir.

## 6.2. Automatització completa de l'estació de treball

Per realitzar el control del nivell del tanc caldrà fer un sistema de regulació, per aquest motiu implementarem un regulador PID.

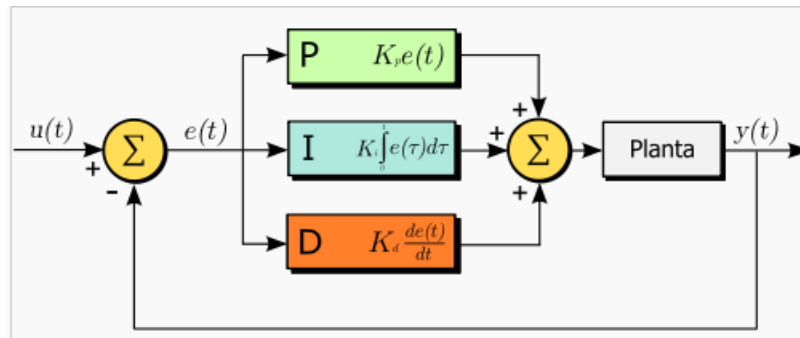


Figura 71. Diagrama de blocs d'un PID

La Figura 71 mostra el diagrama de blocs d'un PID, a on:

- PV o Valor Present de la variable a controlar. Normalment sol ser una entrada analògica, en el nostre cas serà el sensor de nivell analògic LT16.
- SP o Punt de Consigna. És el que volem aconseguir. És una variable d'entrada des d'un terminal HMI o un SCADA, el nivell que nosaltres desitgem.
- MV, Variable Manipulada, en el nostre cas es el senyal de control enviat a la motobomba P1.

Un PID és un mecanisme de control per realimentació que calcula la desviació o error entre un valor mesurat i el valor que volem obtenir, per aplicar una acció correctora que ajusti el procés.

En l'algoritme de càlcul del control PID s'utilitzen tres paràmetres diferents: la proporcional, l'integral, i el derivatiu.

L'error és la desviació existent entre el punt de mesura i el valor consigna, o "Punt de Consigna" SP.

## Proporcional

El valor Proporcional determina la reacció de l'error actual.

La part proporcional consisteix en el producte entre el senyal d'error i la constant proporcional per aconseguir que l'error en estat estacionari s'aproximi a zero, com es pot veure a l'Equació 1.

$$P_{sal} = K_p e(t) \quad (\text{Equació 1})$$

## Integral

L'Integral genera una correcció proporcional a la integral de l'error, això ens assegura que aplicant un esforç de control suficient l'error de per l'acció proporcional es redueix a zero.

El valor de control Integral té com a propòsit disminuir i eliminar l'error en estat estacionari, provocat per l'acció proporcional.

El control integral actua quan hi ha una desviació entre la variable i el punt de consigna, integrant aquesta desviació en el temps i sumant-la a l'acció proporcional. L'error és integrat, la qual cosa té la funció de fer la mitja o sumar-ho per un període determinat. La resposta integral és addicionada a l'acció proporcional per formar el control PI amb el propòsit d'obtenir una resposta estable del sistema sense error estacionari. A l'Equació 2 podem veure com es calcula l'acció Integral.

$$I_{\text{sal}} = K_i \int_0^t e(\tau) d\tau \quad (\text{Equació 2})$$

## Derivatiu

El Derivatiu determina la reacció del temps en el qual l'error es produeix.

L'acció derivativa es manifesta quan hi ha un canvi en el valor absolut de l'error (si l'error és constant, només actuen les variables proporcional i integral).

La funció de l'acció derivativa és mantenir l'error al mínim corregint-ho proporcionalment amb la mateixa velocitat que es produeix; d'aquesta manera evita que l'error s'incrementi. Es deriva pel que fa al temps, es multiplica per una constant  $K_d$  i després es suma als senyals anteriors (PI). És important adaptar la resposta de control als canvis en el sistema ja que una major acció derivativa correspon a un canvi més ràpid i el controlador pot respondre correctament. A l'Equació 3 podem veure el càlcul de la Derivada.

$$D_{\text{sal}} = K_d \frac{de}{dt} \quad (\text{Equació 3})$$

La suma d'aquestes tres accions és usada per ajustar el procés via un element de control com la posició d'una vàlvula de control o l'energia subministrada a un escalfador, per exemple.

Ajustant aquestes tres variables en l'algoritme de control del PID, el controlador pot proveir un control dissenyat pel què requereixi el procés a realitzar. La resposta del controlador pot ser descrita en termes de resposta del control davant un error, el grau el qual el controlador arriba a la consigna, i el grau d'oscil·lació del sistema.

### 6.2.1. Aplicació del PID a la nostre estació de treball

L'objectiu de control serà la regulació del nivell del dipòsit ALT.

Per sintonitzar els paràmetres del PID cal tenir un model del procés, en el nostre cas, el model el determinarem de forma experimental. Amb aquest objectiu, aplicarem un canvi tipus graó a la bomba i observarem com varia el nivell del dipòsit.

Per visualitzar i adquirir la lectura del nostre sensor analògic de nivell LT1.6 i la motobomba farem servir la funció trace. La captura de dades s'activa clicant amb el botó dret al nostre treball->open->trace, com mostra la Figura 72.

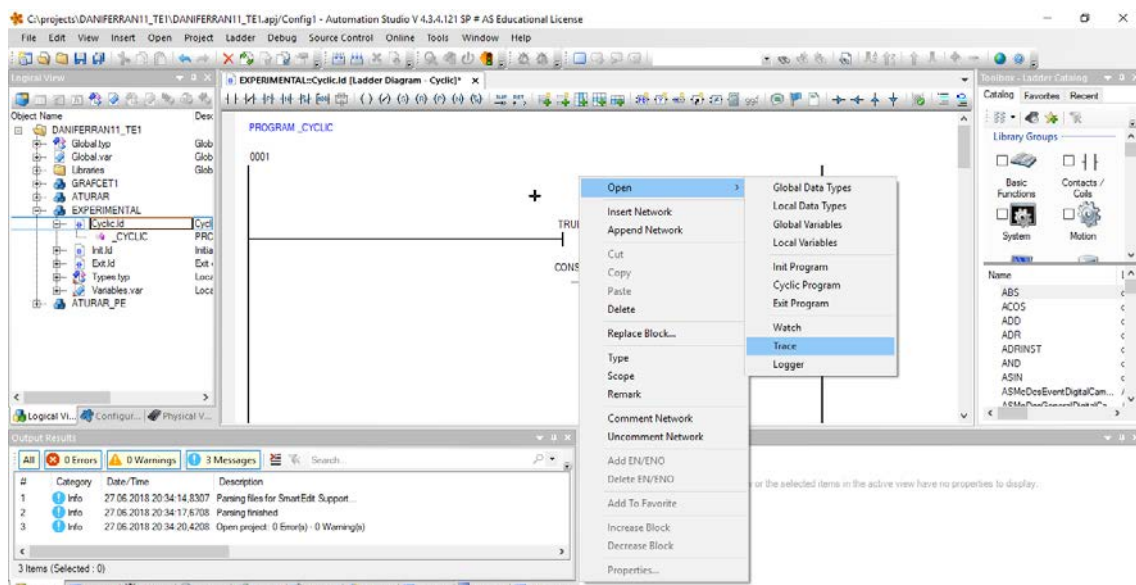


Figura 72. Trace

Un cop desplegat el trace, haurem d'instal·lar-li les variables que volem adquirir, en el nostre cas volem adquirir la lectura del sensor LT1.6 i la tensió aplicada a la motobomba P1.1. Com es veu a Figura 73 clicarem amb el boto dret al trace i a insert variable, i seleccionarem el sensor i la motobomba.

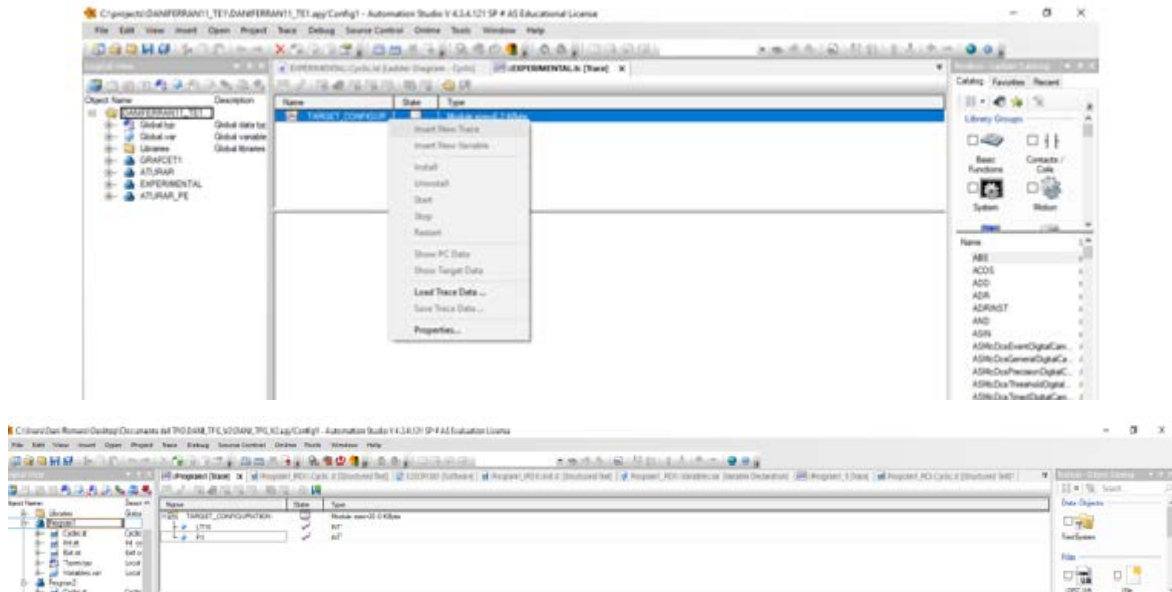


Figura 73. Inserir variable al trace

Després d'agregar les variables, clicarem amb el botó dret al trace i després a install, per integrar-lo al nostre programa. Ja estem preparats per graficar les dades. Posarem en marxa el nostre programa, li posarem una consigna a la nostra motobomba, en el nostre cas hem assignat un valor de 16000 amb un MOVE, posarem a true V1.4 i amb un MOVE obrirem al màxim l'electrovàlvula V1.5.

Un cop el senyal que llegim, és estable, podem considerar que el nivell del tanc s'ha estabilitzat i donarem més potència a la bomba, en concret hem augmentat el valor a 25000. Quan la lectura del sensor s'hagi tornat a estabilitzar pararem el trace amb stop. Seguidament guardarem la gràfica amb format excel, per fer-ho clicarem, com ens mostra la Figura 72, a Load trace data i ho guardarem en format excel.



Figura 74. Guardar gràfica del trace

Les dades del excel les passarem al programa MATLAB i traurem la gràfica, com es pot veure a la Figura 75, trobarem el valor inicial quan el sensor s'estabilitza i a la Figura 74, es veu el valor final.

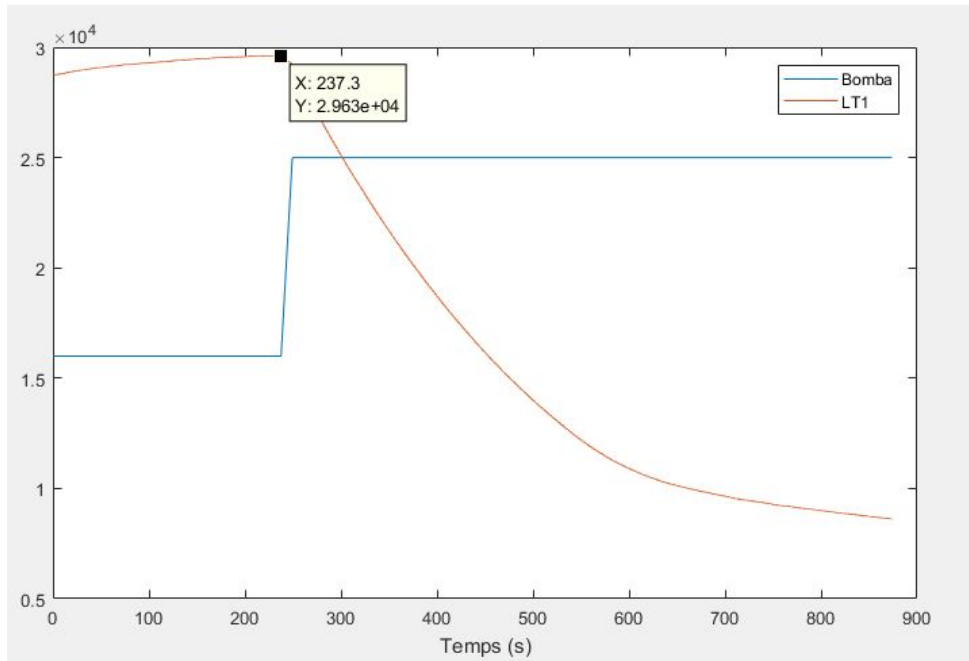


Figura 75. Valor inicial

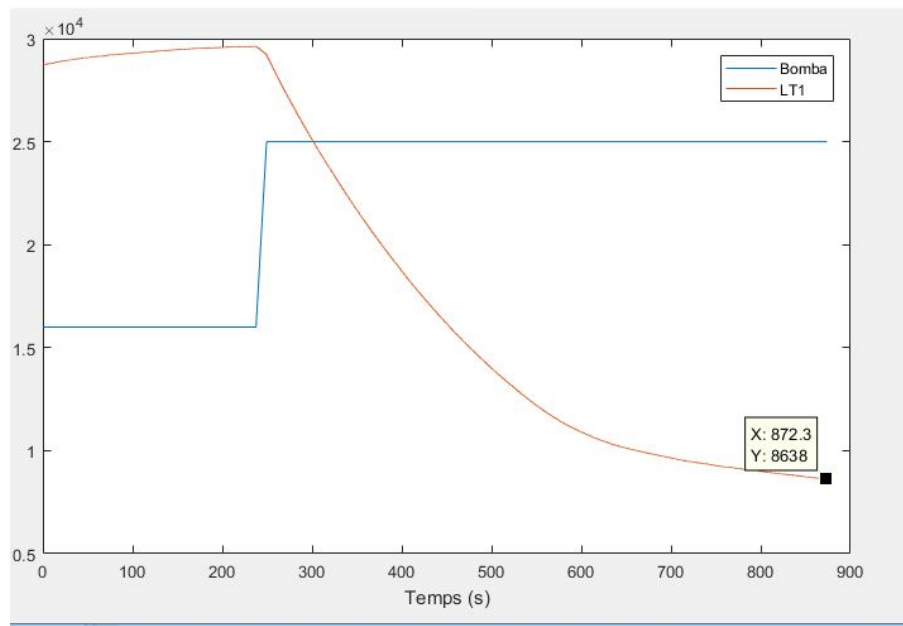


Figura 76. Valor final.

Amb aquestes dades determinem el model del sistema ajustant-lo a un sistema de primer ordre amb un guany i una constant de temps. Com veiem en primer lloc calcularem el guany del sistema:

$$Guany = \frac{(valor\ final - valor\ inicial)_{del\ sensor\ LT\ 1.6}}{(valor\ final - valor\ inicial)_{de\ la\ motobomba}} = \frac{(8638 - 29630)}{(25000 - 16000)} = 2,33$$

El valor de la constant de temps l'hauré de calcular de la diferència entre el valor inicial i el valor final de la lectura del sensor analògic i trobar el temps en aquest punt.

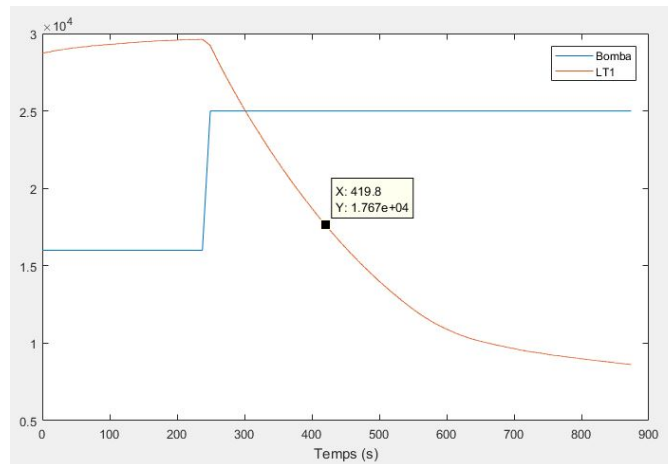


Figura 77. Càlcul del 63% de la diferència entre el valor final i el inicial

La constant de temps es calcula mirant el temps que triga el procés a arribar al 63,2% del seu valor final. Per tant com es veu a la Figura 77 el temps en aquest punt des del punt inicial a aquest punt es més o menys de 270 s.

Emprant aquests valors i aplicant la tècnica de sintonia s'han calculat uns paràmetres pel regulador PID. A l'implementar les constants calculades, s'ha observat que el sistema de control no tenia un bon comportament i s'ha optat per fer servir la funció d'auto sintonia (autotune).

Aquesta funció està localitzada a la finestra de function blocs i s'anomena MTBascisStepTunnig. Ens calcularà les nostres constants proporcional, constant de temps Integral, constant de temps derivada i la constant de temps del filtre de la (derivada Figura 78).



MTBasicsStep Tuning	
EN	ENO
Enable	Active
System Settling Time	Error
Max Tuning Time	StatusID
MinActValue	UpdateDone
MaxActValue	Out
Update	TuningActive
ActValue	TuningDone
StepHeight	PIDParameters
Start	Quality

Figura 78. MTBascisStepTunnig

Les variables que li hem d'entrar són:

- Posar sempre a true l'enable del PID (EN). Enable, és l'habilitació de les nostres variables introduïdes al PID. Fins que no estigui a true no habilitarà les variables, per tant no començarà l'auto Tunning. Si durant l'auto Tunning canviem qualsevol variable caldrà fer un reset a Enable.
- El System Setting Time, que serà el nostre temps de cicle del programa, en aquest cas és de 100 ms que haurem d'entrar-li en segons, 0,1 s.
- Max Tunning Time que serà una mica més de la constant Integral calculada. En aquest cas 250 s.
- Min Actual Value, serà el valor mínim que pot llegir el nostre sensor analògic on volem aplicar el PID. En el nostre cas el nostre sensor LT 1.6 pot llegir de 0 a 32600.

- Max Actual Value, serà el valor màxim que pot llegir el sensor. En el nostre cas és 32600.

A Act Value introduïrem el valor que llegeix el nostre sensor LT 1.6, però li haurem d'entrar el valor de la variable amb tipus REAL, no Integer. Per tant haurem de fer una conversió d'Integer a Real com mostra Figura 79.

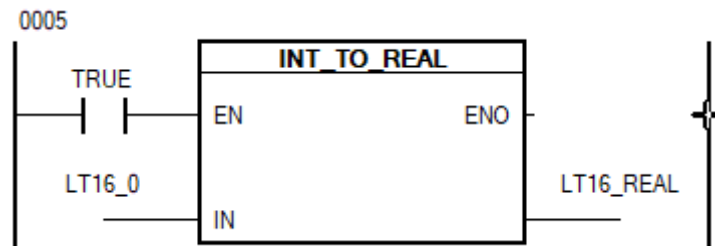


Figura 79. Conversió de INTEGER a REAL

- Step Heigh: serà el nostre graó, que en el nostre cas farem que el graó de la bomba sigui de 20000.
- Start: l'haurem de posar a true quan vulguem començar a realitzar l'auto Tuning.
- A la sortida OUT és on volem que actuï el PID, en el nostre cas a la motobomba. Com que ens retorna un valor REAL i el tipus de variable de la motobomba és Integer, haurem de realitzar una conversió de REAL a INTEGER, com mostra la Figura 80.

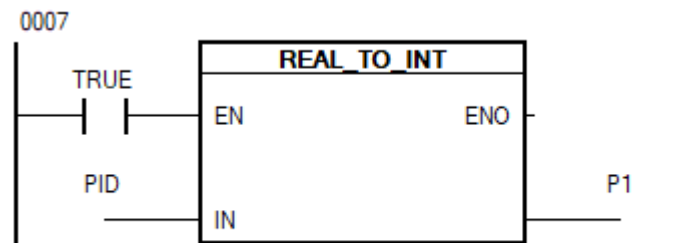


Figura 80. Conversió de REAL a INTEGER

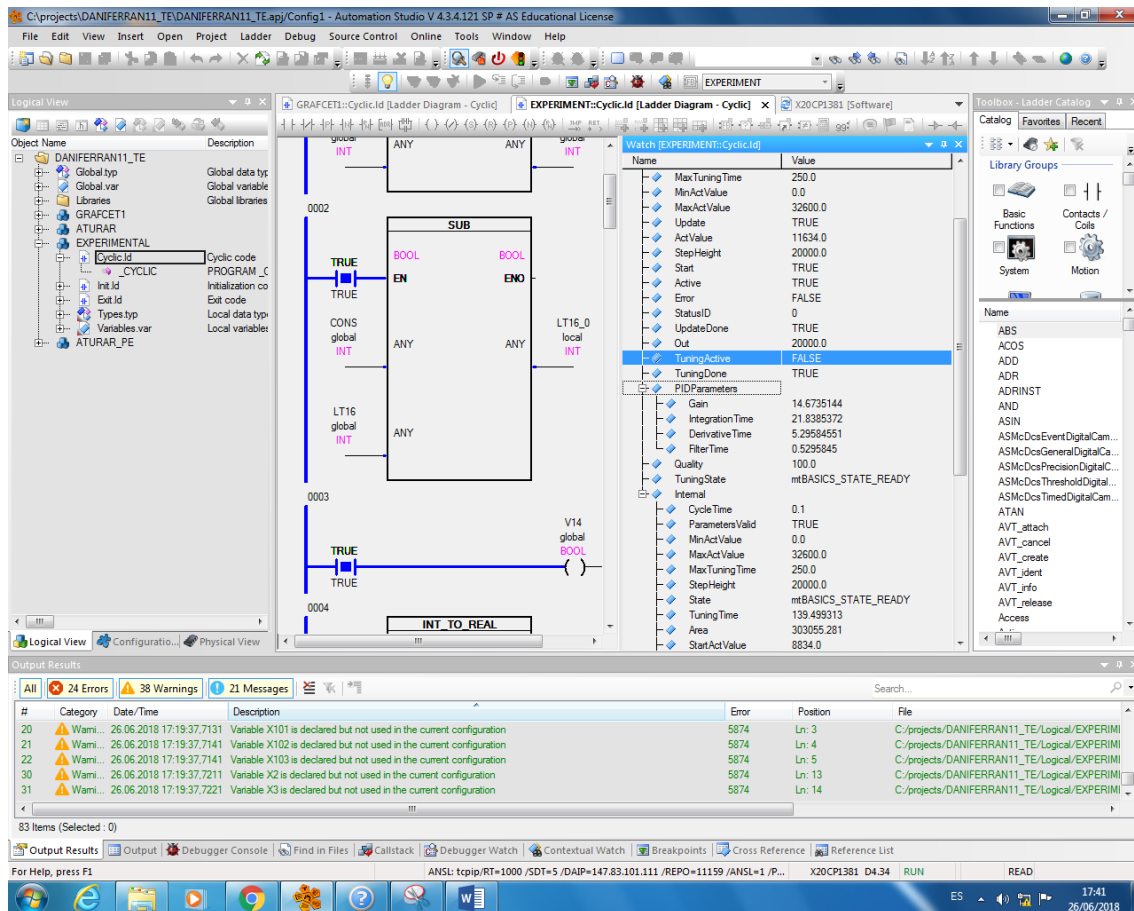


Figura 81. Constants PID

Per començar a realitzar l'auto Tune, haurem de passar l'Enable de false a true, tenir Start a True, i TuningActive a true. Llavors la funció ens realitzarà el càlcul de les constants com es pot veure a Figura 81.

Un cop ens hagi retornat les constants del nostre sistema, passarem a implementar el PID al nostre programa. El PID el trobarem a la llibreria FB (Function Blocks), i seleccionarem el MTBASICS\_PID. Els valors que li haurem d'entrar són els que ens ha retornat l'auto TUN.

- Gain = 14,6735144 (Kp)
- Integration Time = 21,838572s
- Derivative Time = 5,29584551
- Filter Time = 0,529845

També li haurem d'entrar a SetValue la consigna que desitgem de nivell, a ActValue serà la lectura del nostre sensor de nivell analògic LT 1.6, el Min i MaxOut que serà el valor mínim i màxim que li podem aportar a la motobomba, a l'Out serà el valor REAL que al fer la conversió, com avanç hem explicat, atacarà a la motobomba i haurem de posar a True

tant l'EN del PID com l'enable de les variables entrades. A la Figura 82 ens mostra l'estructura del PID. També haurem de declarar el MTBasicsPID com a variable local del nostre programa. Hem de recordar que totes les variables que el PID tracta han d'estar en el format que demana, fent les conversions pertinents com hem explicat anteriorment. Per exemple l'OUT és tipus REAL, haurem de realitzar una conversió a INTEGER ja que és el format en el que treballa la motobomba.

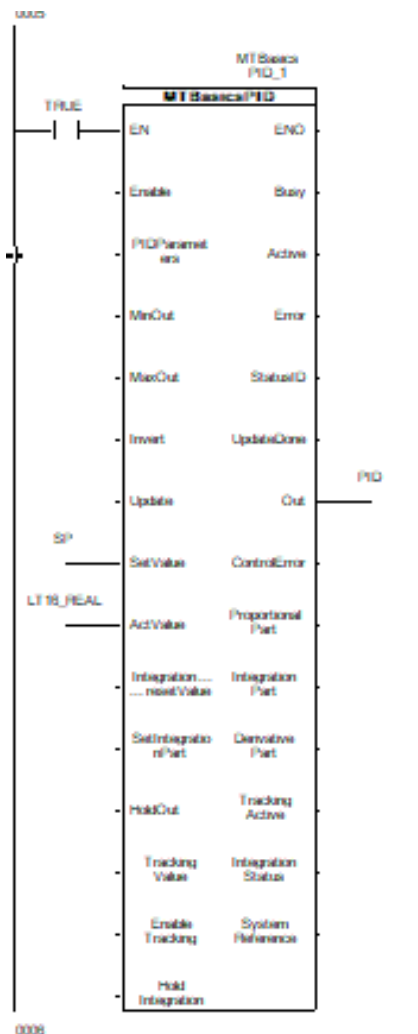


Figura 82. Function Blocks PID

Un cop configurat, provarem si en el nostre sistema enllaç tancat funciona correctament abans d'implementar-lo al programa. En el nostre cas l'acció derivativa i el filtre, ens fa que quan el nivell està a punt d'arribar a la consigna, la motobomba comenci a oscil·lar molt el seu valor. Per tant considerem que l'acció derivativa i el filtre són prescindibles, ja que al provar-ho sense aquestes constants de temps, el sistema respon millor sense oscil·lacions a l'arribar a la consigna.



El projecte on implementarem el PID, serà on hem implementat tots els programes anteriors i funcionarà de la forma que es descriu seguidament,

Perquè la seqüència comenci haurem prémer el botó START, i que el tanc inferior assoleixi el nivell mínim. Un cop es compleixin aquestes condicions, obrirem l'electrovàlvula V1.4 i posarem en marxa el temporitzador de 2 segons. Quan hagin transcorregut aquests 2 segons, posarem en marxa la bomba i començarem a transportar l'aigua del tanc inferior al tanc superior. Un cop estigui funcionant la motobomba esperarem que el nivell inferior del tanc inferior ens marqui que té el nivell mínim. Llavors és quan ens actuarà el PID, fins que el nivell LT1.6, que és la variable a regular, arribi a la consigna. Un cop hagi arribat a la consigna activarem un temporitzador de 7 segons. Quan acabi el temps del temporitzador passarem a la següent etapa que atura el PID, per tant, s'aturarà la motobomba, i obrirem l'electrovàlvula V 1.5, que farà que el tanc superior comenci a buidar-se traspasant l'aigua cap al tanc inferior fins que el valor del nivell analògic sigui igual o més gran a la de la variable N0 (nivell inicial). I així successivament a no ser que polsem a ATURA o al PE, que realitzaran les mateixes funcions que en els programes anteriors.

A la Figura 83 ens mostra la implementació del nostre PID al Grafcet de PE.

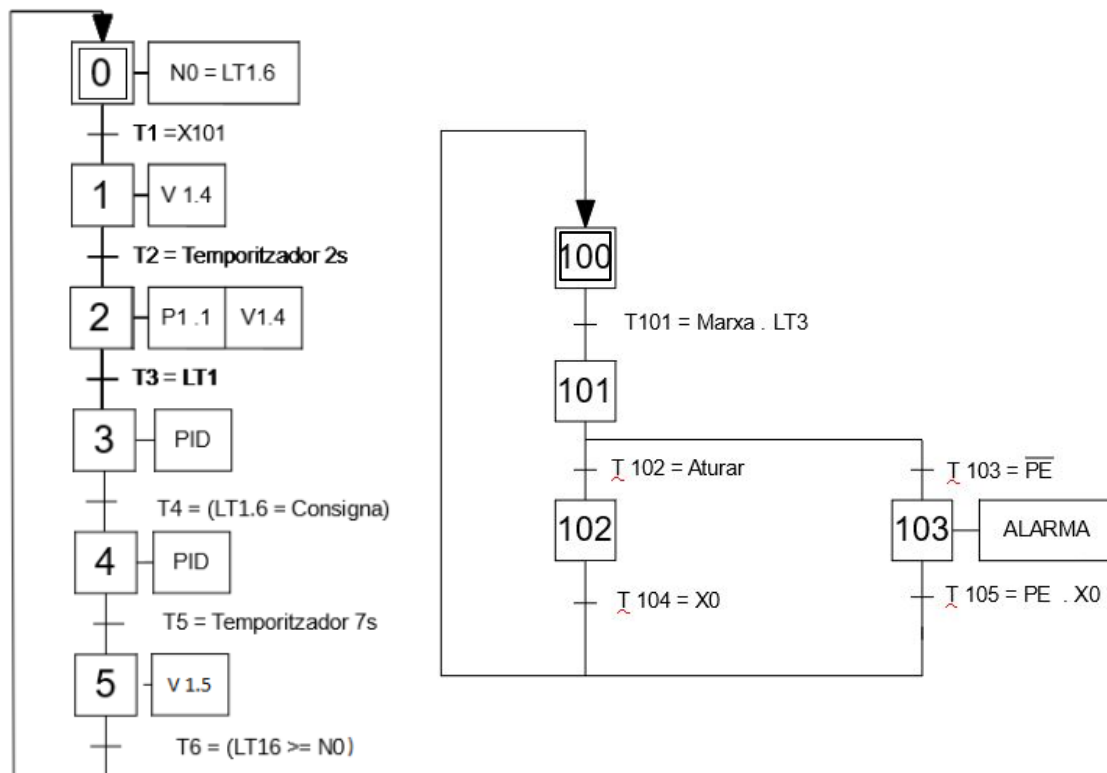


Figura 83. Grafcet PID

El programa el trobarem al Annex 1 i s'anomena PID.

## 7. REGULACIÓ DEL NIVELL I MONITORITZACIÓ

En aquest apartat farem servir la monitorització virtual del programa Visu. El primer que farem serà incloure en el nostre projecte la pantalla amb la que voldrem comandar la nostra estació. A la logical view-> boto dret-> add object (de la mateixa manera que incloem un programa) visual components i triem la VC4 com es pot veure a la Figura 84.

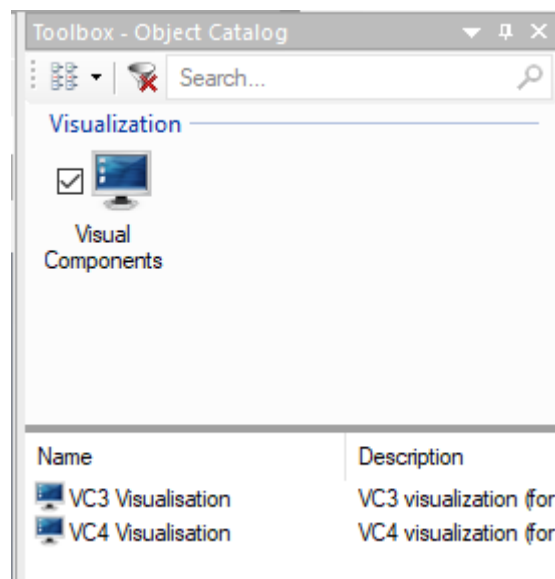


Figura 84. Add Visu

El següent pas serà començar a confeccionar la nostra pantalla. Obrirem la Visu, fent doble clic a sobre de Visu i obrirem la finestra init page com es veu a la Figura 83.

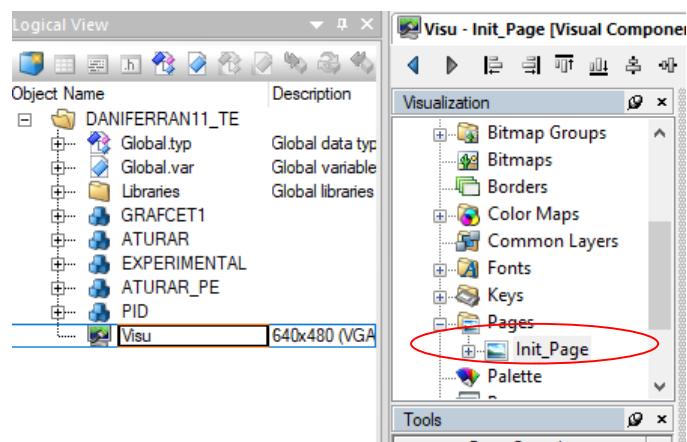


Figura 85. Init Page

Ara ja podem incloure a la pantalla tots els components que vulguem per comandar la nostra pantalla. En el nostre cas inclourem:

- 3 Buttons, que seran per START, ATURAR i PE.
- 1 visualitzador numèric del sensor LT 1.6
- 1 entrada numèrica per posar el valor de la consigna de nivell.
- 2 LEDs que ens marcaran si els nivells LT1 i LT3 tenen nivell.

Per incloure qualsevol element a la pantalla haurem de seleccionar l'element que desitgem i arrossegar-lo fins a la pantalla. En tots els elements hi ha la possibilitat de canviar les seves característiques físiques, com donar-li una forma, canviar el color, lletra, etc... Ho podem veure a la Figura 86.

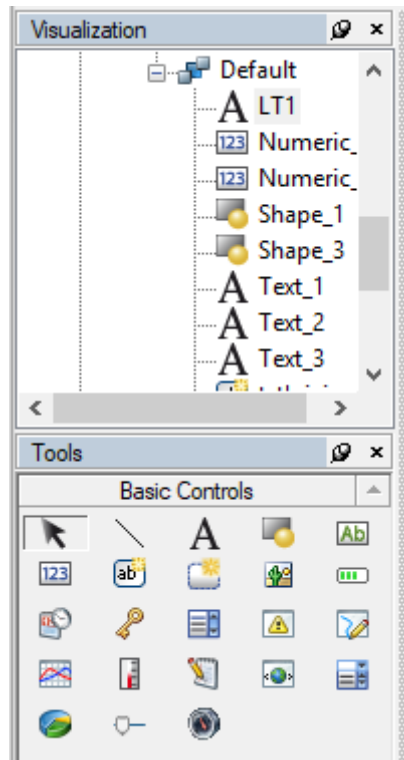


Figura 86. Elements del Visu

Per associar el botó START amb la nostra variable farem el següent:

- Posar-li nom perquè aparegui al botó. Obrirem dins de les opcions del botó, Text Source-> Text-> i a English li donarem el nom.
- Per associar el botó a la variable START, Keys-> Action-> Value-> Data Point i clicarem a Set Data Point i allà l'associarem a la variable que vulguem, en aquest START.

A la Figura 87 podem veure totes les opcions del button.



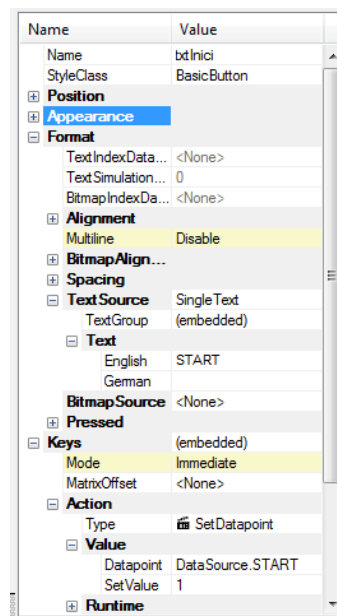


Figura 87. Configuració Button

Per als botons d'Aturar i de PE realitzarem els mateixos passos.

Per al visualitzador numèric de LT1.6, escollirem l'element numèric i a Style class cal dir-li que és un Input, a Value -> Data Point i a Set Data Point li associarem la variable de LT1.6 com es pot veure a la Figura 88.

Per a l'entrada numèrica de la consigna serà el mateix procediment, però serà un Output i l'associarem a la variable SP que serà la nostra consigna de nivell com es pot veure a la Figura 89.

S'ha de tenir en compte que els elements Numeric, només poden associar-los a variables tipus Integer.

Name	Value
Name	Numeric_1
StyleClass	Input
<b>Position</b>	
<b>Appearance</b>	
Border	Flat_black
ForeColor	0
BackColor	15
ColorMap	<None>
ColorDatapoint	<None>
Font	Arial9pxValue
<b>Format</b>	
<b>Alignment</b>	
MinIntegerDigits	1
AddFractionDigits	0
AddFractionDigit...	<None>
<b>Unit Text</b>	None
<b>Value</b>	Standard
Datapoint	DataSource.PID.LT16_0
MinValue	0
MinDatapoint	<None>
MaxValue	32670
MaxDatapoint	<None>
TeachDatapoint	<None>
MinMaxViolation	Limit to Min/Max
SimulationValue	0
<b>Input</b>	True
<b>Runtime</b>	
ControlID	0

**Appearance.BackColor**  
Selects the object's background color.

Figura 88. Configuració visualitzador numèric

Name	Value
Name	Numeric_2
StyleClass	Output
<b>Position</b>	
<b>Appearance</b>	
Border	Flat_grey
ForeColor	255
BackColor	15
ColorMap	<None>
ColorDatapoint	<None>
Font	Arial9pxValue
<b>Format</b>	
<b>Alignment</b>	
MinIntegerDigits	1
AddFractionDigits	0
AddFractionDigit...	<None>
<b>Unit Text</b>	None
<b>Value</b>	Standard
Datapoint	DataSource.PID.SP_INT
MinValue	0
MinDatapoint	<None>
MaxValue	32600
MaxDatapoint	<None>
TeachDatapoint	<None>
MinMaxViolation	Limit to Min/Max
SimulationValue	0
<b>Input</b>	True
<b>Runtime</b>	
ControlID	0

**Appearance.BackColor**  
Selects the object's background color.

Figura 89. Configuració entrada numèrica

Per visualitzar els sensors analògics de nivell farem servir l'element Shape. El primer que haurem de fer serà configurar-li un color altrament no deixa associar-lo a una variable booleana. Obrirem color maps i Add color Map, després li configurarem els colors que vulguem (Figura 90).

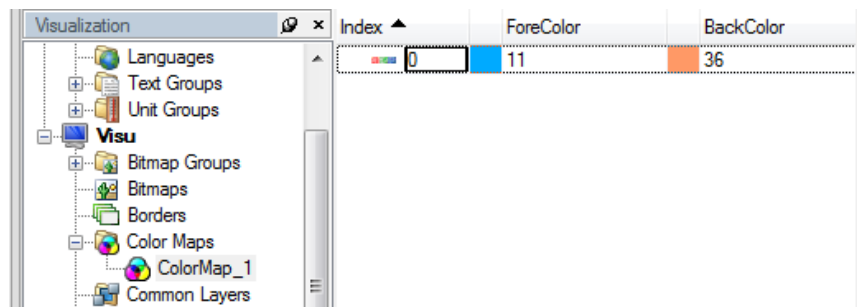


Figura 90. Color Maps

Una vegada configurat el color map anirem a les opcions del Shape, i al color Data Point ja podrem associar-li la variable booleana que vulguem, en aquest cas LT1 es pot veure a la Figura 91.

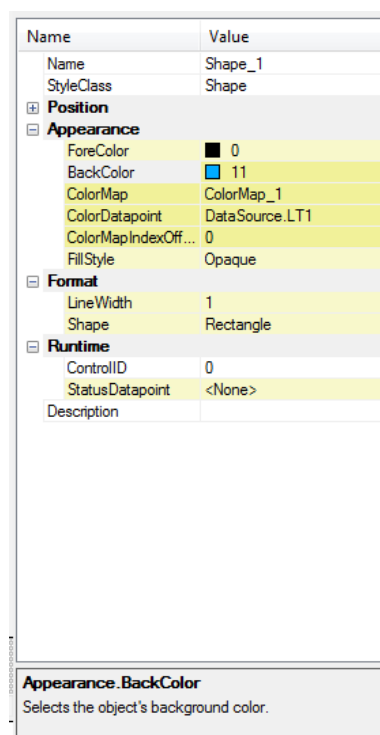


Figura 91. Configuració del shape

La pantalla quedarà de la següent manera.( Figura 92.)

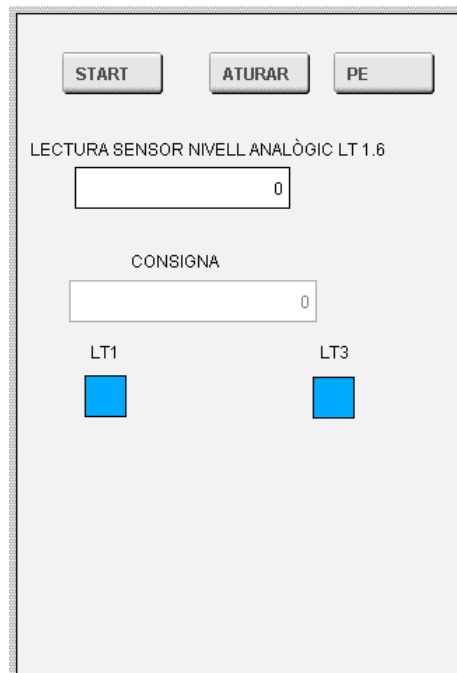


Figura 92. Pantalla final

El següent que farem serà configurar la Visu dins del nostre projecte per poder connectar-nos virtualment. Anirem a Physical view-> ETH-> Configuration, dins de configuration a VC mapping-> VC Object Name li associarem el nostre Visu, com es pot veure a la Figura 93.

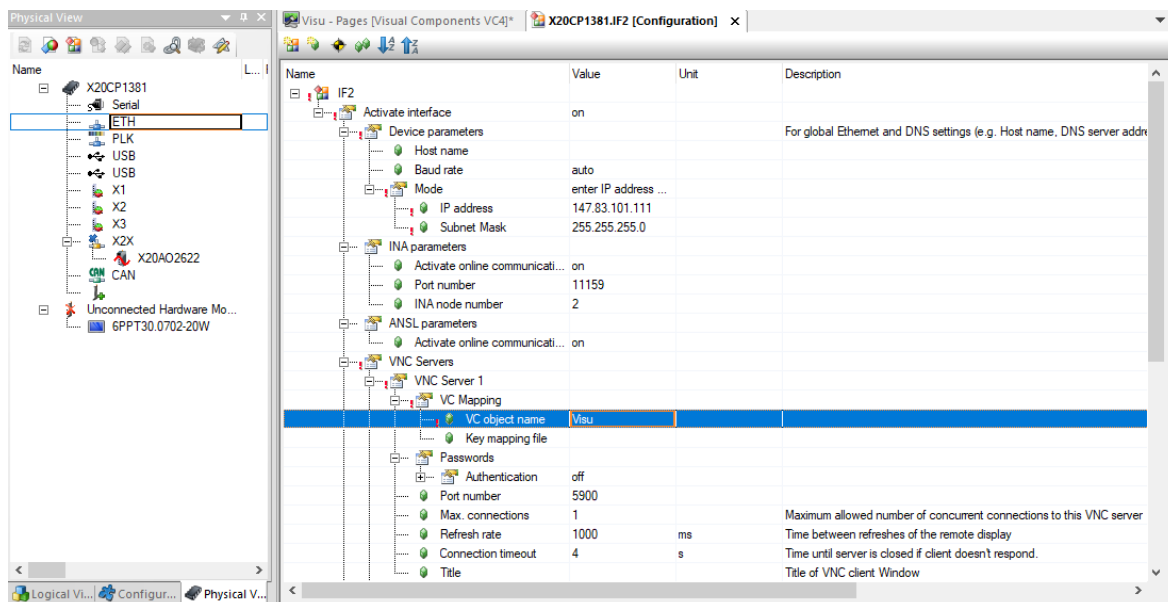


Figura 93. Configurar Visu

El següent que farem serà instal·lar-nos el programa VNC Viewer. Un cop ens hem instal·lat el programa, compilarem i transferirem el programa i la Visu al PLC. Un cop estigui transferit

el programa i el PLC estigui en RUN, obrirem el VNC viewer i l'únic que haurem de fer serà posar la IP del PLC i el port ho veiem a la Figura 94.

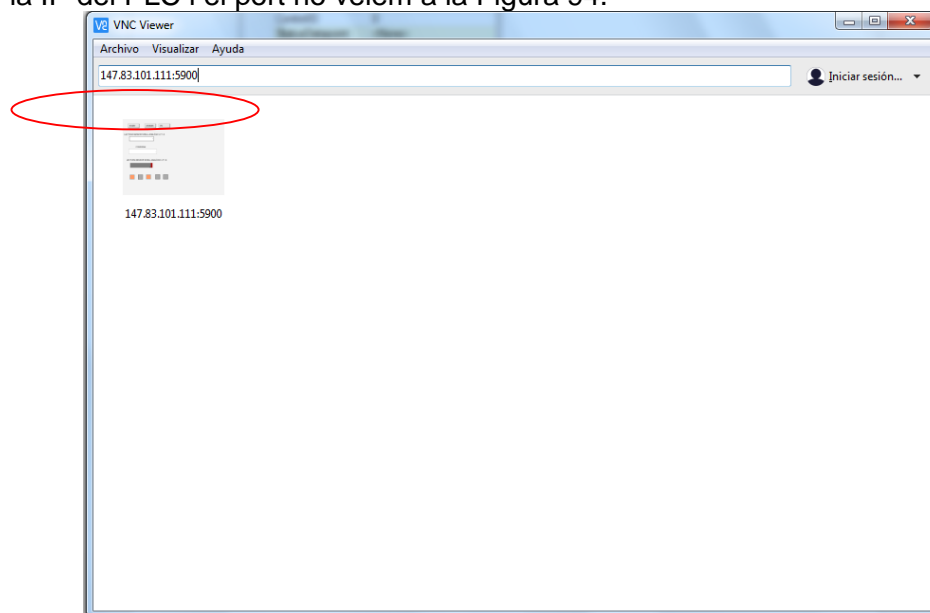


Figura 94. VNC viewer

Automàticament se'ns obrirà la nostra pantalla virtual i ja podrem començar a controlar el nostre sistema des d'aquesta. Es pot veure com queda la pantalla a la Figura 95.

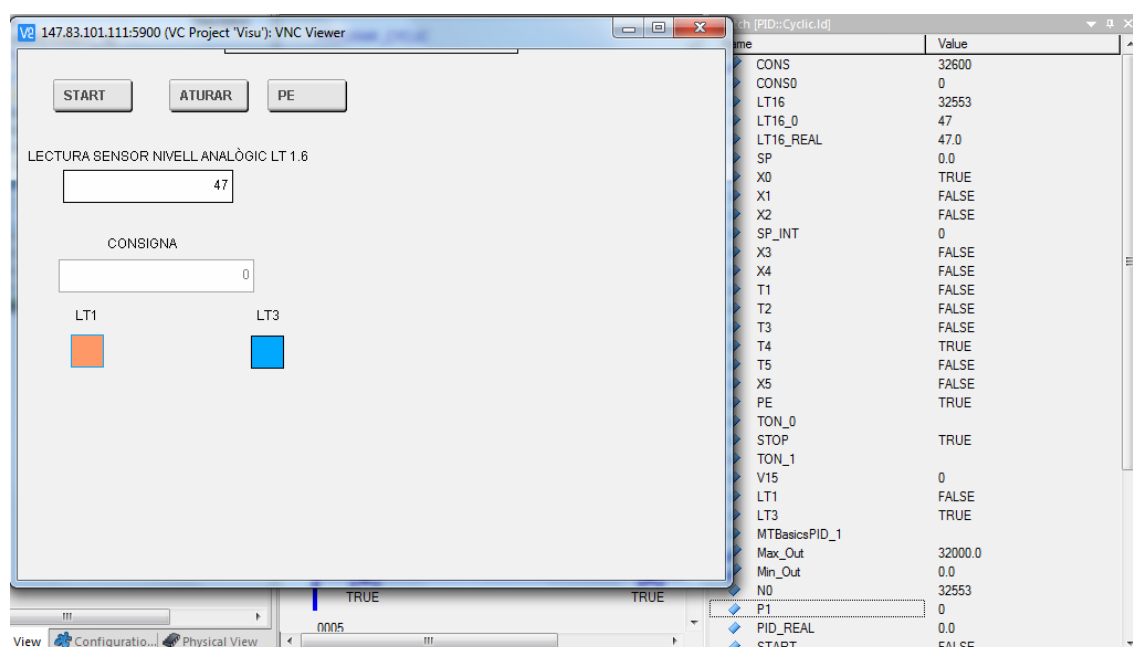


Figura 95. Pantalla virtual online



## 8. PROVES, PROBLEMES I SOLUCIONS

Un cop anava implementant els programes, els transferia al PLC i comprovava el correcte funcionament respecte del funcionament teòric, connectant-me online (funció watch) veia en tot moment el valor de les variables, estat de les etapes, transicions, electrovàlvules, motobomba, sensors, etc... No vaig tenir cap complicació en aquest aspecte.

Les complicacions més importants amb les que ens hem trobat, han sigut a l'hora d'implementar el PID ja que hi ha molt poca informació de l'Automation Studio (internet), però amb les ajudes trobades al mateix Atomation Studio vam arribar a trobar l'autotune del PID i un cop trobades el mateix FB ens va retornar les constants del nostre sistema per implementar el PID.

El problema més important amb el que ens hem trobat ha sigut a la recta final del treball, quan volíem implementar la pantalla que ens van donar B&R, però al final no s'ha pogut fer servir, ja que no estava preparada per ser connectada amb la nostre CPU, cosa que en un principi pensàvem que sí. En un futur seria aconsellable realitzar la programació i connexió d'aquesta perquè la interacció home-màquina sigui més completa.

I on vam perdre més temps va ser en el disseny i fabricació de les 3 estacions, ja que es va haver de fer des de 0 tot el muntatge.



## 9. CONCLUSIONS

Amb conclusió crec que ha sigut molt gratificant tot el procés de disseny, fabricació i automatització de l'estació, el coneixement de l'entorn de programació de l'Automation Studio.

He confeccionat una guia bàsica per poder seguir fàcilment per programar amb el PLC B&R que és bastant completa, posant diferents programes com a exemple.

Aquestes maquetes serien de molta utilitat per a qualsevol indústria de tractament de líquids, ja que es podrien realitzar proves per a futures innovacions a petita escala, per després portar-ho a terme a la realitat a la indústria. Jo vaig fer les pràctiques en una empresa de tractament de lixiviats, amb diferents processos automatitzats amb diferents tancs on es realitzaven diferents mescles de diferents líquids. Aquestes estacions serien d'una gran importància per aquest tipus d'indústria, en la meva opinió.







## 10. BIBLIOGRAFIA

Clare, W., Kaplan, G., Saldon, D., Wiktorowicz, A., Gilbert, R., & Wendt, C. (2006). *PLCs: Programmable Logic Controllers*.

Morley, D. (2018). *History of the PLC*. <https://library.automationdirect.com/history-of-the-plc/>: eBook, Lybrary.AutomationDirect.

Segovia, V., & Theorin, A. (2012). History of Control History of PLC and DCS. *University of Lund*.

Vyatkin, V. (2009). The IEC 61499 standard and its semantics. *IEEE Industrial Electronics Magazine*, 3, 4, 40-48.


























## 11. ANNEXOS























### 11.1. Annex 1

## 1 Logical View

Object Name	Description
DANIFERRAN11_TE	
Global.typ	Global data types
Global.var	Global variables
Libraries	Global libraries
operator	This library contains function interfaces for IEC 61131-3 operator functions. For the most part, these are mathematical and logical functions.
operator.fun	
operator.typ	
operator.var	
runtime	This library contains runtime functions for IEC tasks.
runtime.fun	
runtime.typ	
runtime.var	
asctime	The AsTime Library supports DATE_AND_TIME and TIME data types.
asctime.fun	
asctime.typ	
asctime.var	
AslecCon	This library contains function interfaces for IEC 61131-3 conversion functions.
AslecCon.fun	
AslecCon.typ	
AslecCon.var	
LoopCont	This library contains control algorithms.
LoopCont.var	
LoopCont.typ	
LoopCont.fun	
sys_lib	The SYS_LIB library contains functions for memory management and operating system manipulation as well as hardware-specific functions.
sys_lib.fun	
sys_lib.typ	
sys_lib.var	
brsystem	The BRSystem library provides the user with a number of system functions for e.g. handling permanent memory, accessing exception information, etc.
brsystem.fun	
brsystem.typ	
brsystem.var	
standard	This library contains standard function blocks and functions for IEC 61131-3.
standard.fun	
standard.typ	
standard.var	
MTBasics	This mechatronics library contains basic control function blocks.
MTBasics.fun	
MTBasics.typ	
MTBasics.var	
MTTypes	This mechatronics library contains different system datatypes and general error numbers.
MTTypes.fun	
MTTypes.typ	
MTTypes.var	
GRAFCET1	
Cyclic.Id	Cyclic code
Init.Id	Initialization code
Exit.Id	Exit code
Types.typ	Local data types
Variables.var	Local variables
ATURAR	
Cyclic.Id	Cyclic code
Init.Id	Initialization code
Exit.Id	Exit code

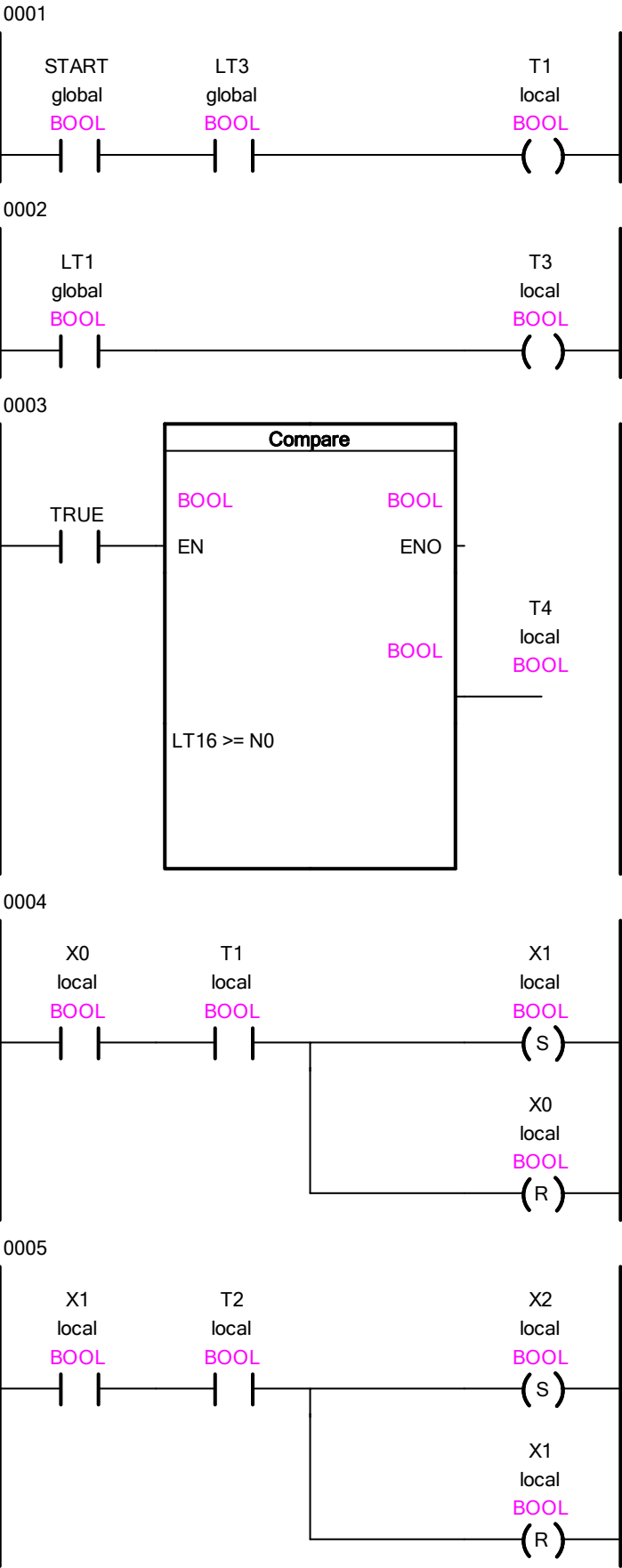
Object Name	Description
 Types.typ	Local data types
 Variables.var	Local variables
 EXPERIMENTAL	
 Cyclic.Id	Cyclic code
 Init.Id	Initialization code
 Exit.Id	Exit code
 Types.typ	Local data types
 Variables.var	Local variables
 ATURAR_PE	
 Cyclic.Id	Cyclic code
 Init.Id	Initialization code
 Exit.Id	Exit code
 Types.typ	Local data types
 Variables.var	Local variables
 PID	
 Cyclic.Id	Cyclic code
 Init.Id	Initialization code
 Exit.Id	Exit code
 Types.typ	Local data types
 Variables.var	Local variables
 Visu	640x480 (VGA)

## 1.1 \\Logical\\Global.var

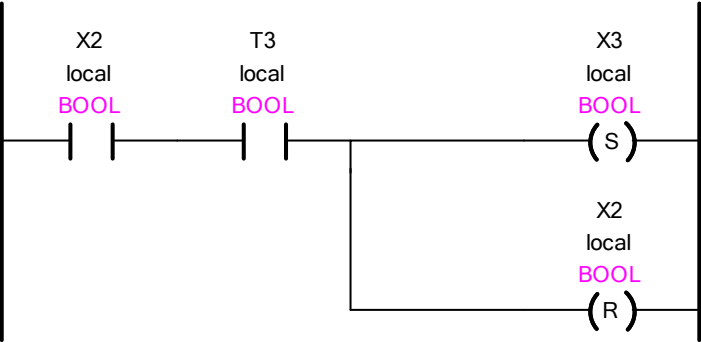
Name	Type	 Constant	 Retain	 Replicable	Value	Description [1]
 CONS0	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
 CONS	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32600	
 LT1	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 LT4	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 LT3	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 START	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 LT2	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 LT5	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 V13	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 V14	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 PT11	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 LT16	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 ALARMA	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 PE	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 V15	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
 V12	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 P1	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
 V11	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
 STOP	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

1.2 \Logical\GRAFCET1\Cyclic.Id

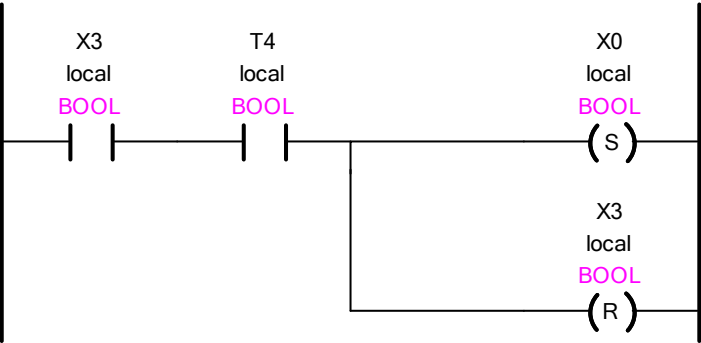
PROGRAM \_CYCLIC



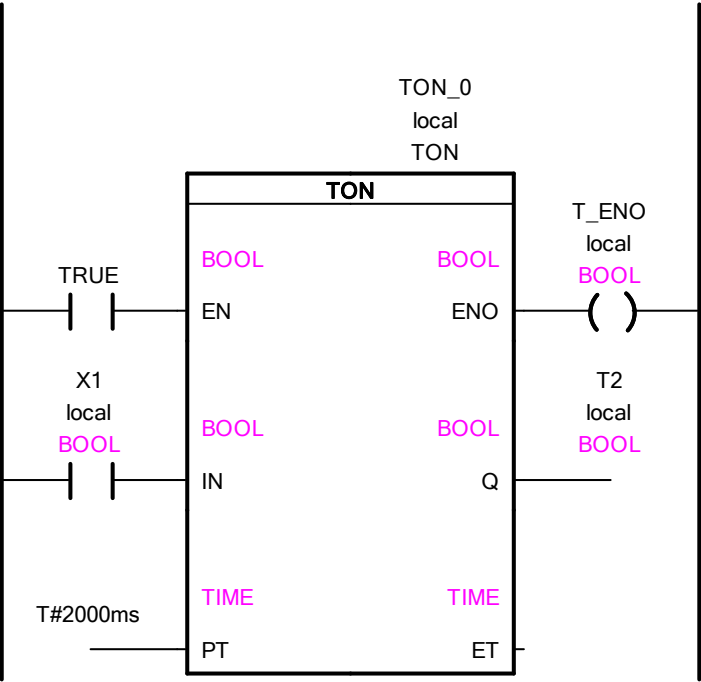
0006



0007



0008

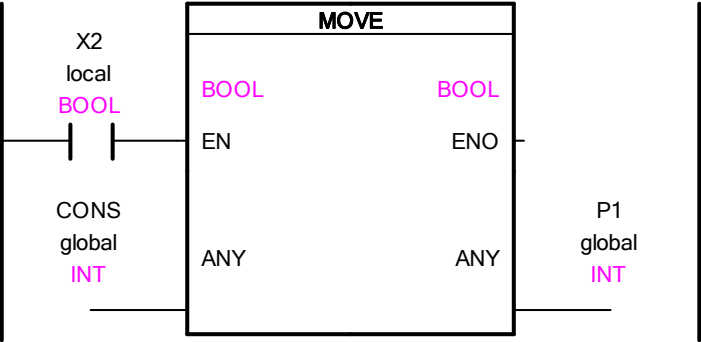


0009

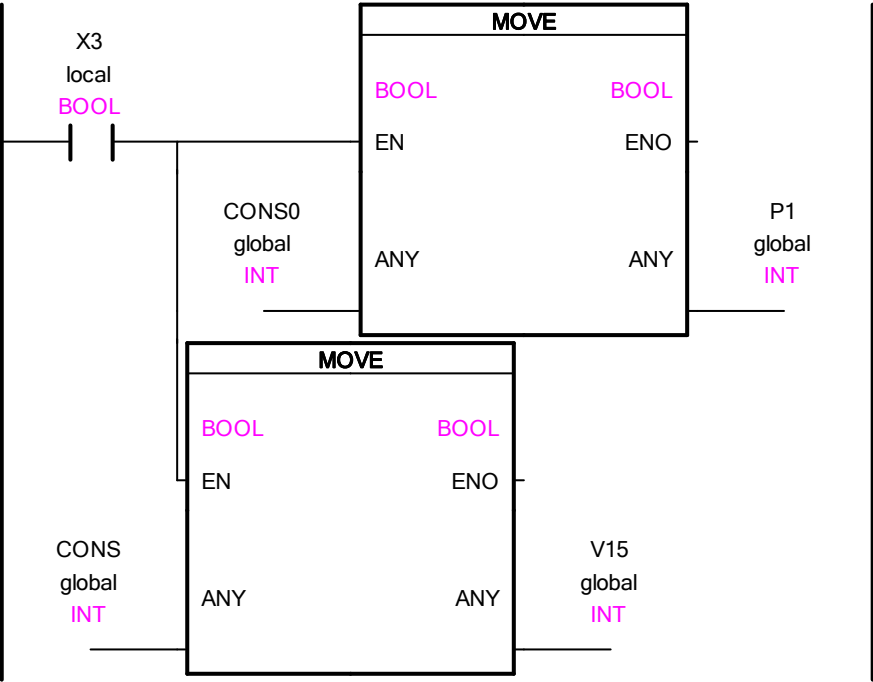




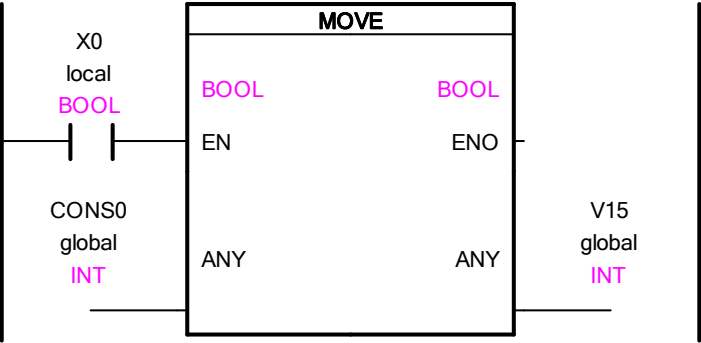
0010



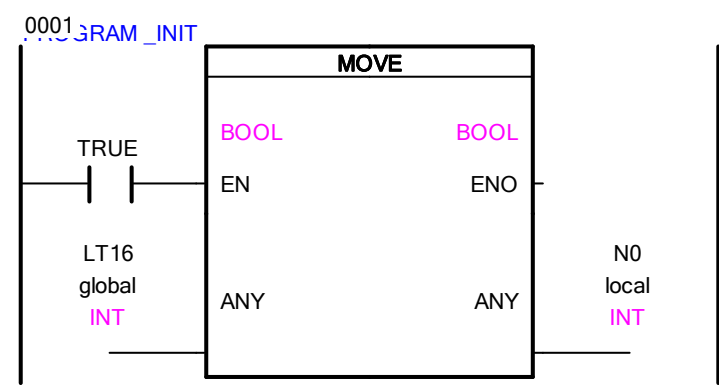
0011



0012



1.3 \Logical\GRAFCET1\Init.Id

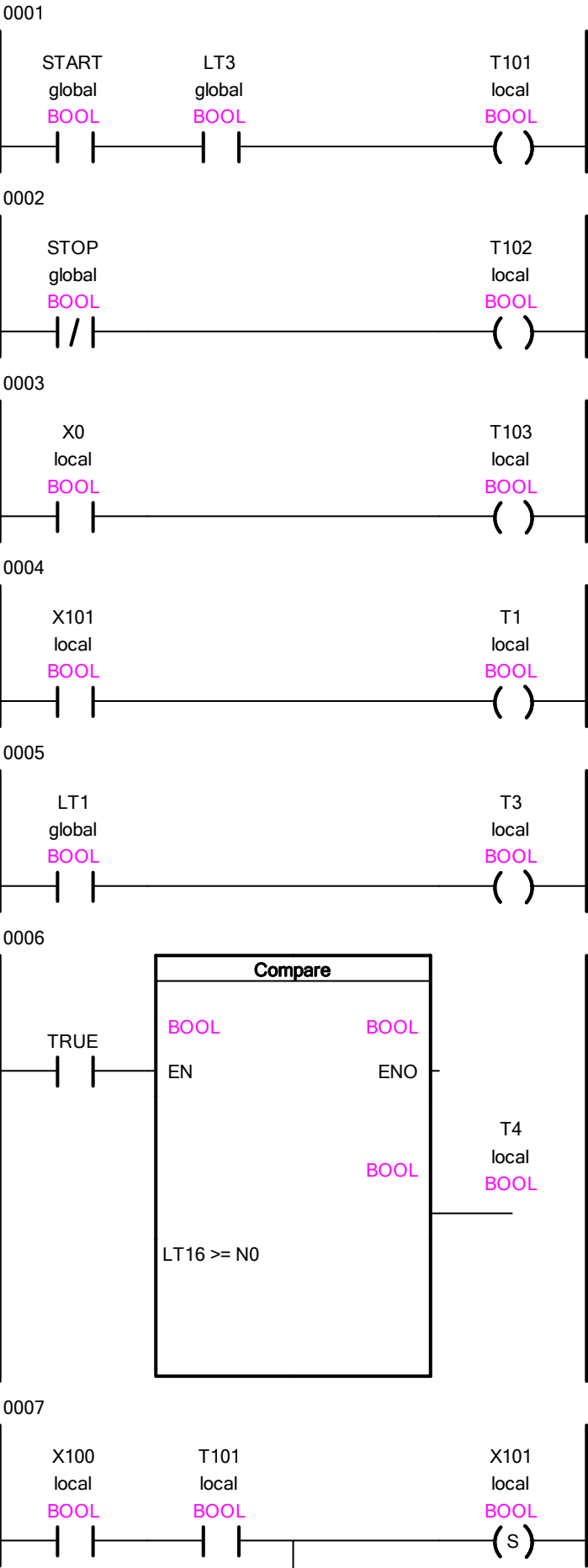


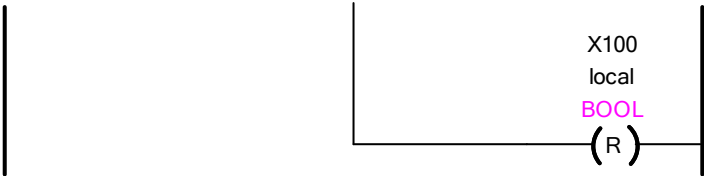
## 1.4 \\Logical\\GRAFCET1\\Variables.var

Name	Reference	Type	Constant	Retention	Replicable	Value	Description [1]
N0	<input type="checkbox"/>	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32500	
V1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
X0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
T1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
T3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
N1	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
BN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
bN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
T4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
X1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
T2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
X2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
X3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
TON_0	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
TIM_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
T_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

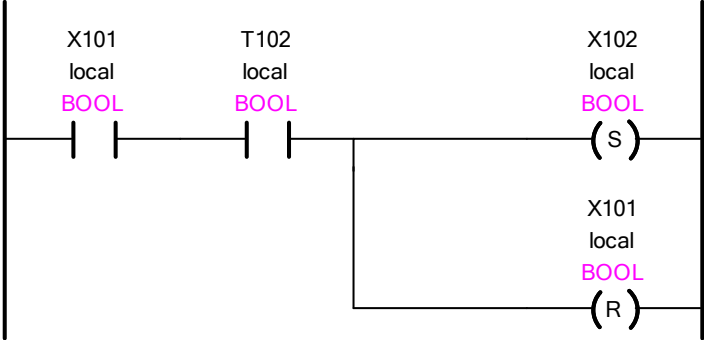
1.5 \Logical\ATURAR\Cyclic.ld

PROGRAM \_CYCLIC

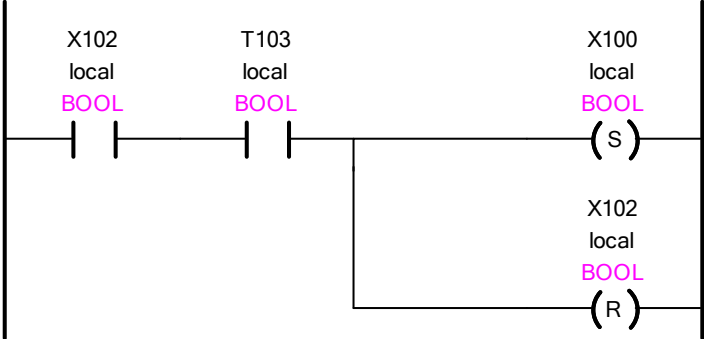




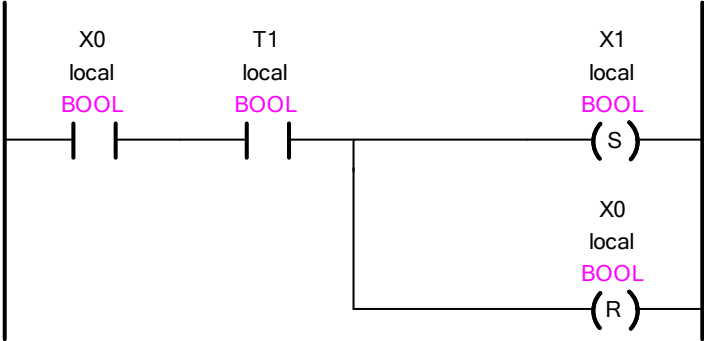
0008



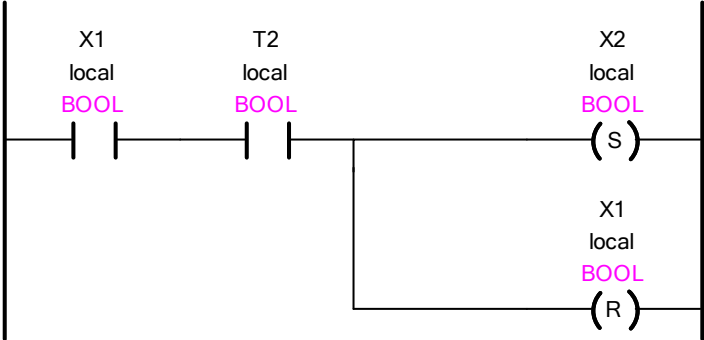
0009



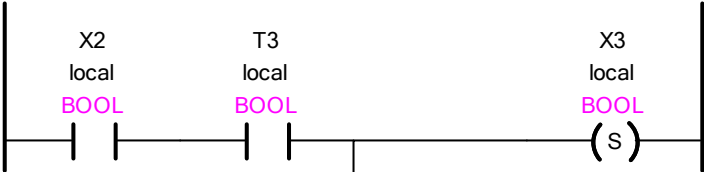
0010

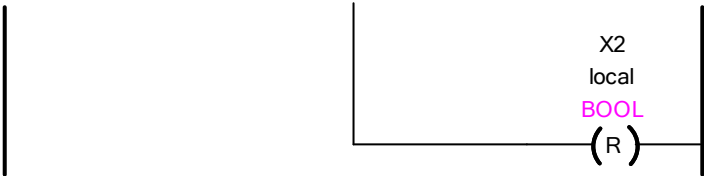


0011

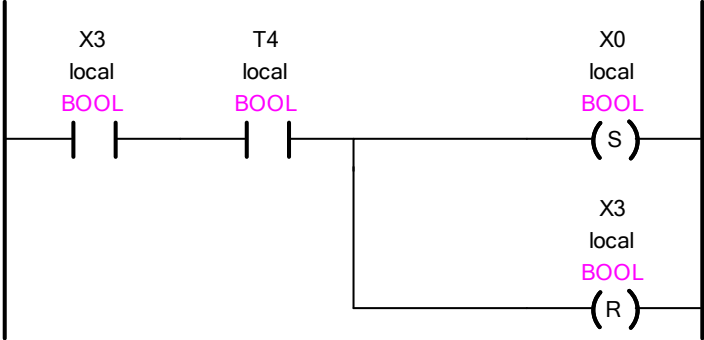


0012

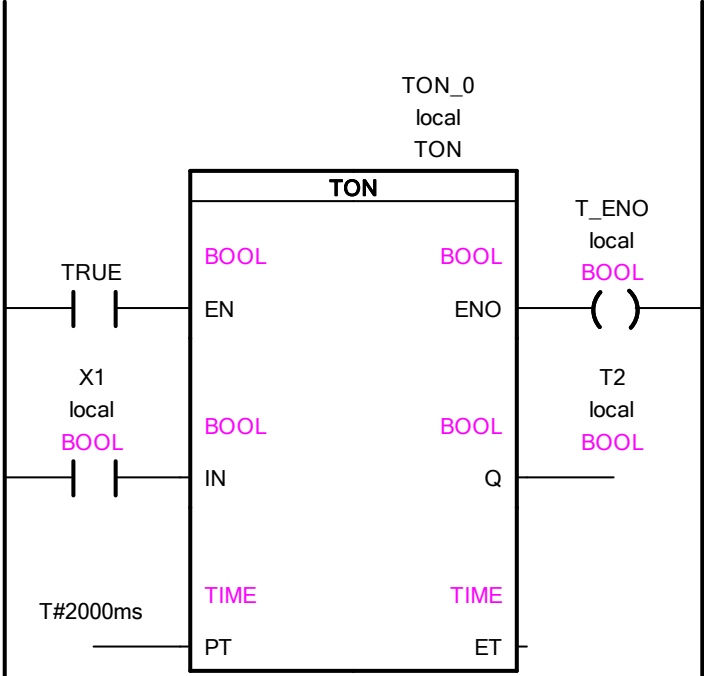




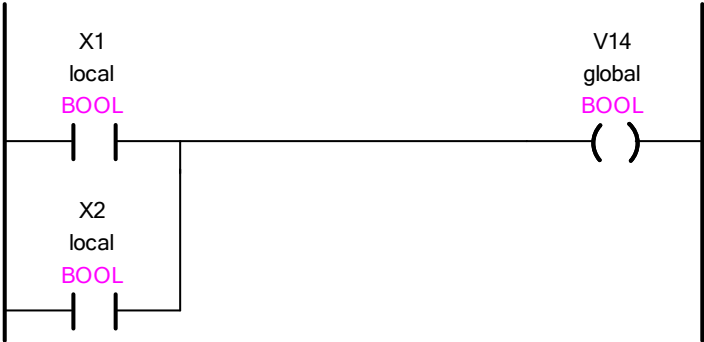
0013



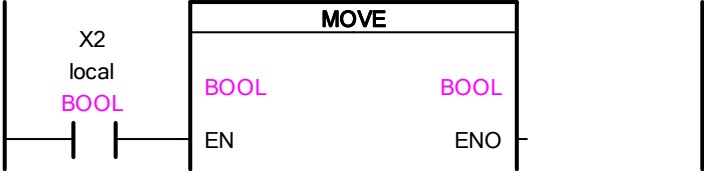
0014



0015

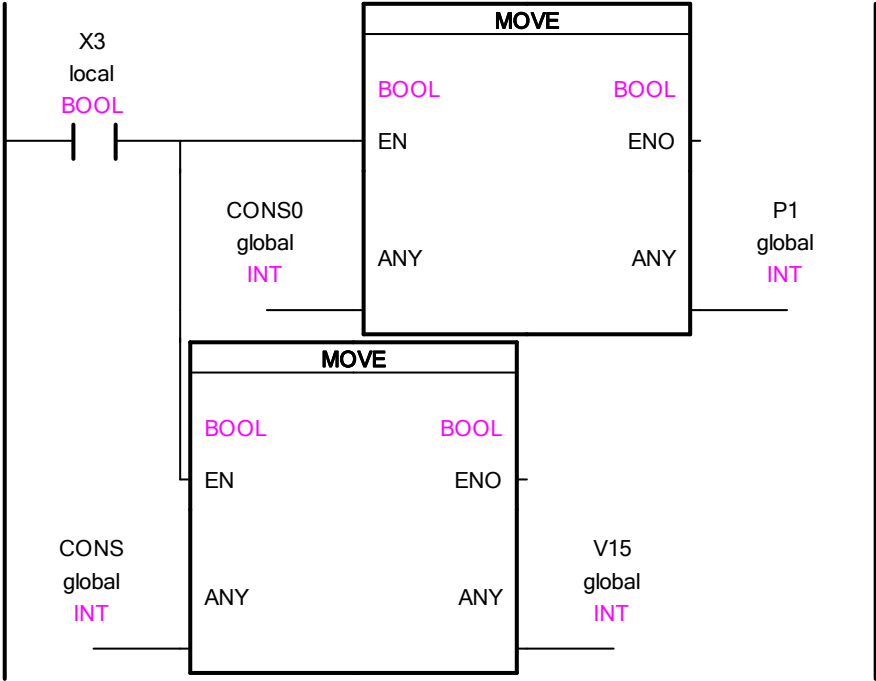


0016

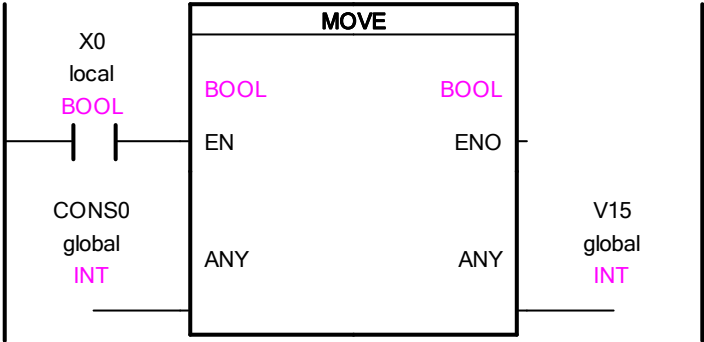




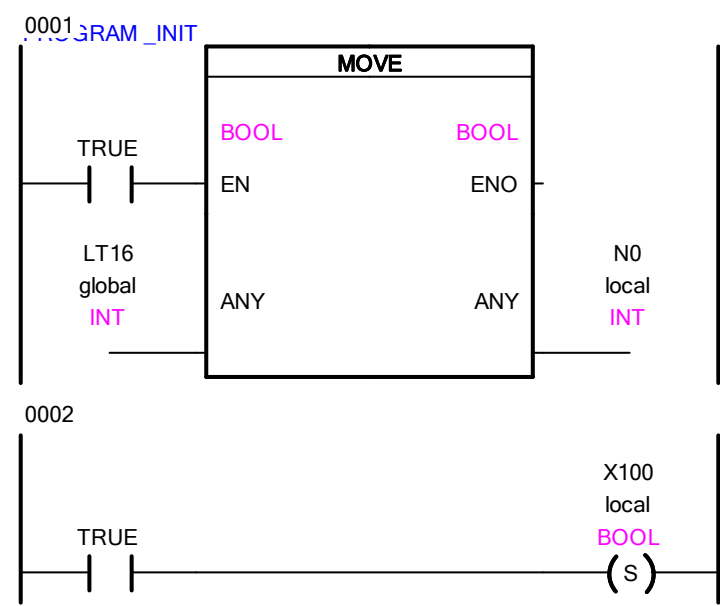
0017



0018

























1.6 \Logical\ATURAR\Init.Id

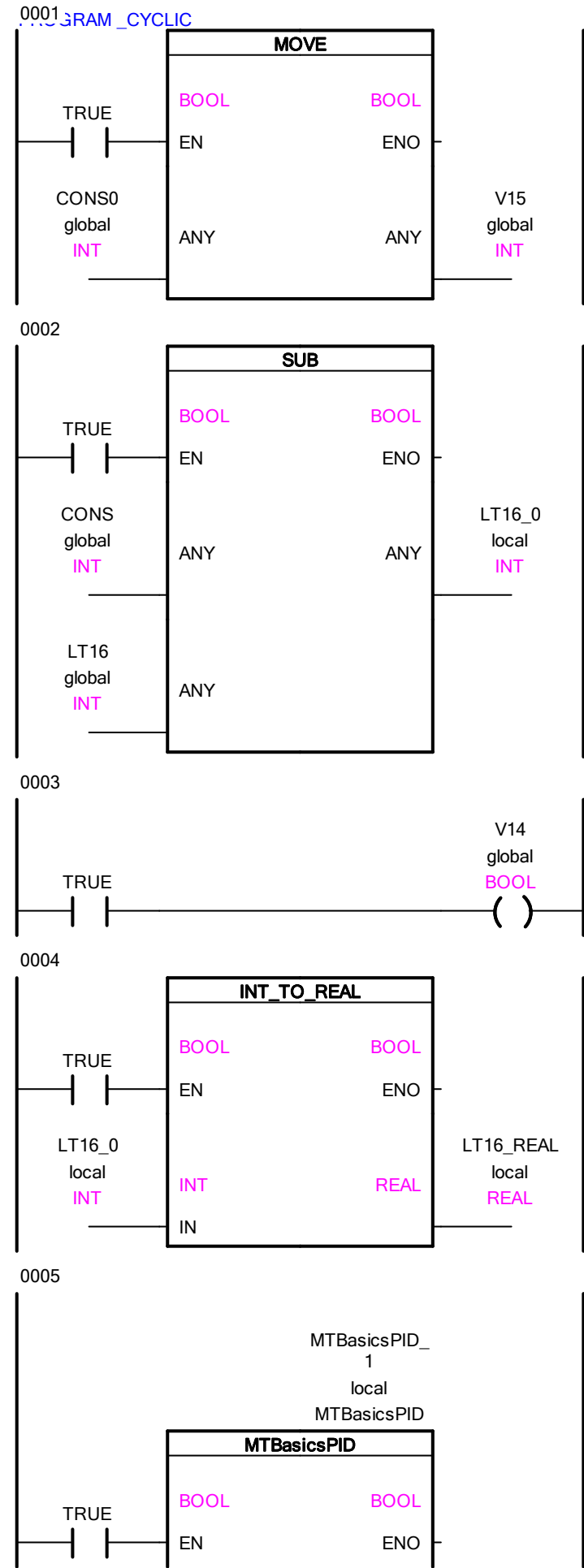


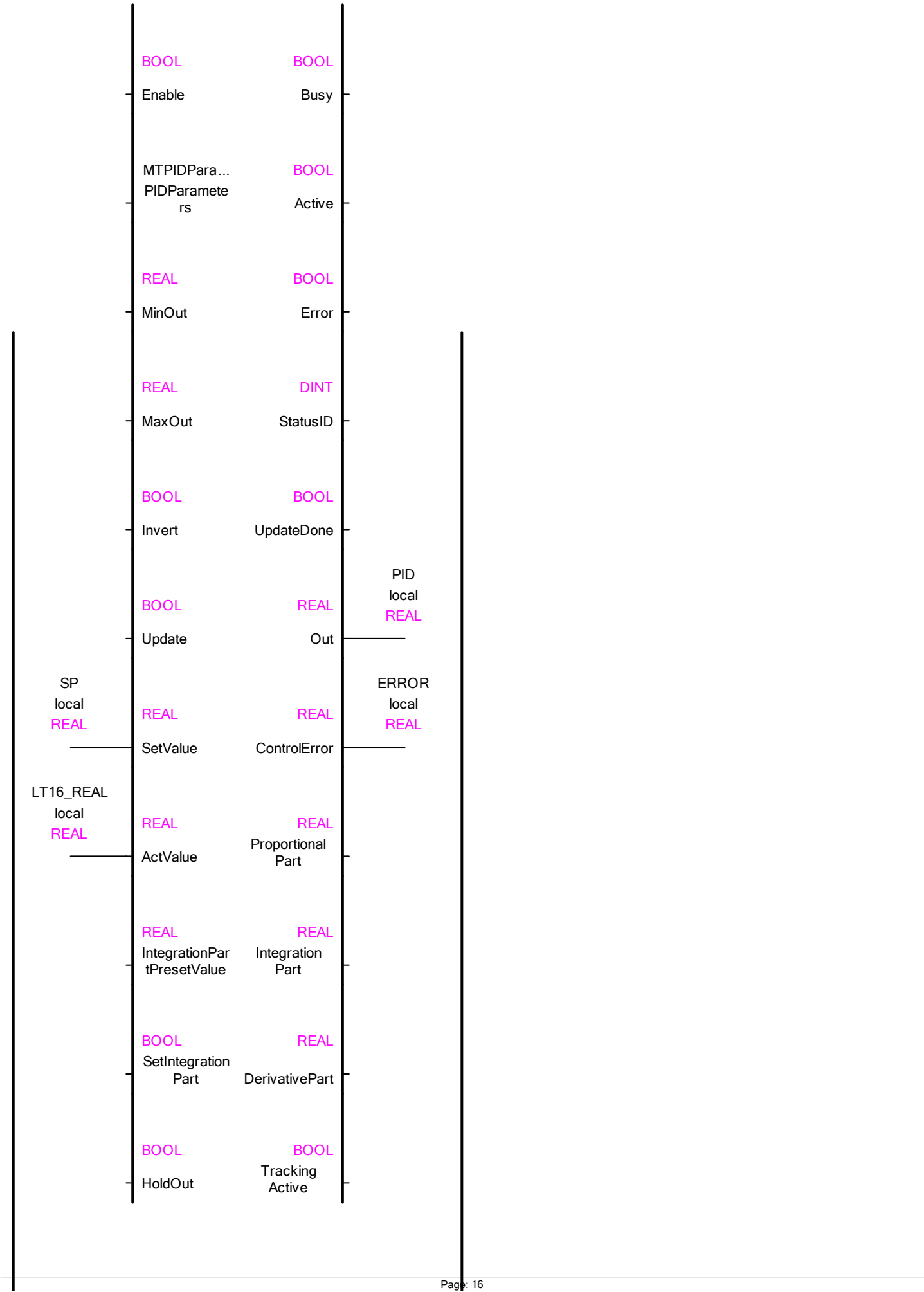


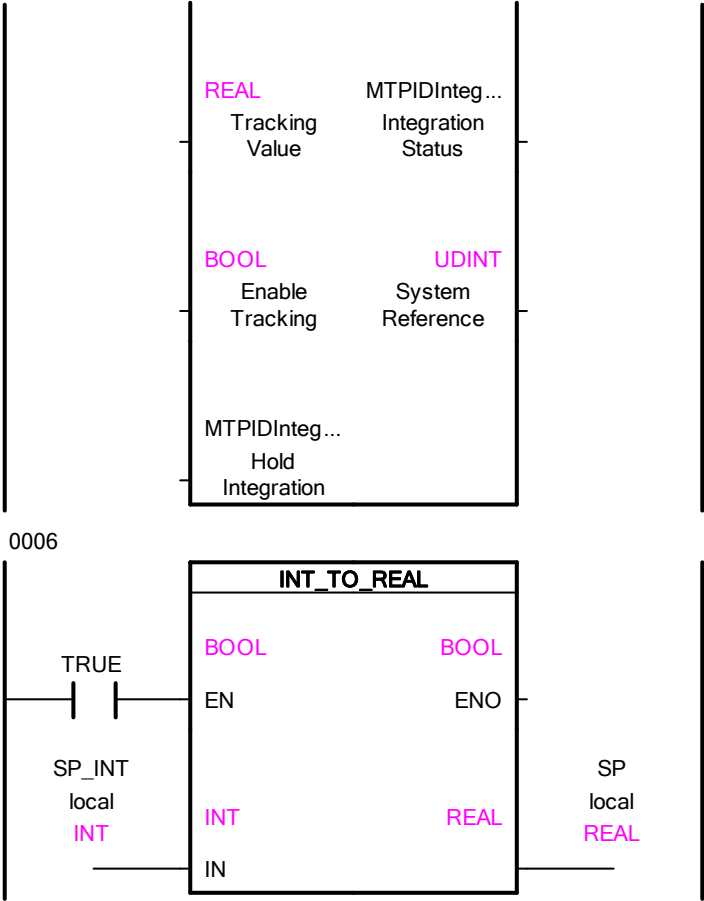
## 1.7 \Logical\ATURAR\Variables.var

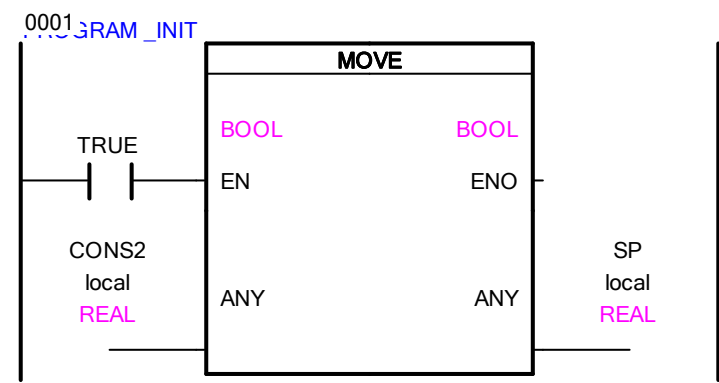
Name	Reference	Type	Constant	Retention	Replicable	Value	Description [1]
 X100	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
 X1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 T1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N1	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N0	<input type="checkbox"/>	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32500	
 TON_0	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TIM_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 BN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 bN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T5	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

1.8 \Logica\EXPERIMENTAL\Cyclic.Id










































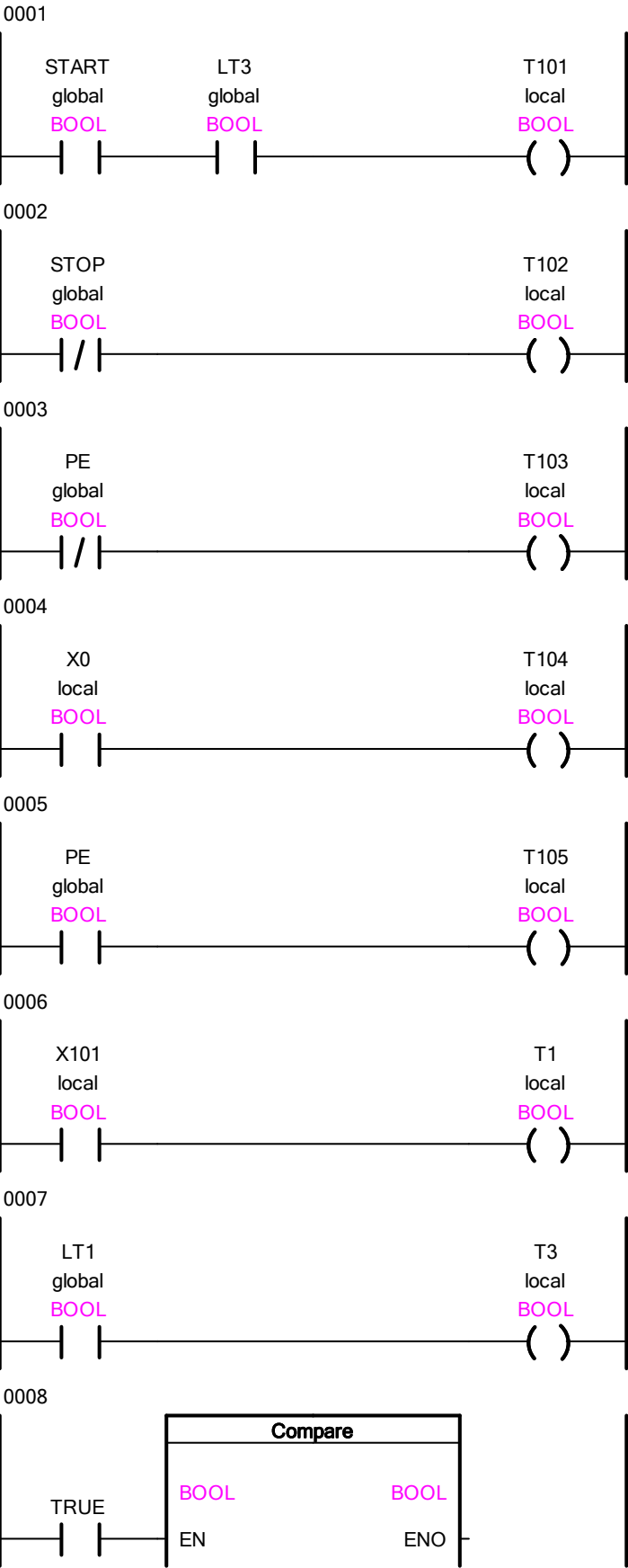


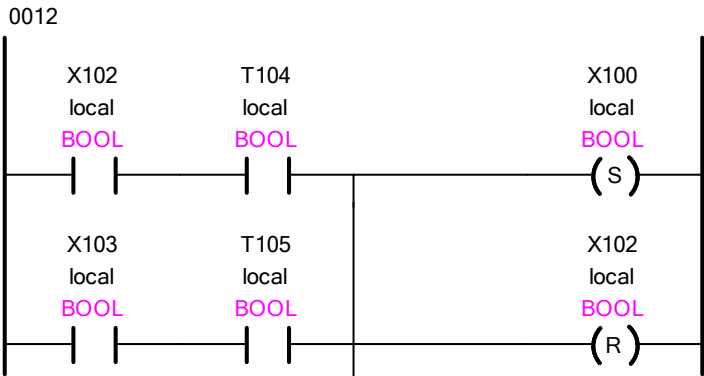
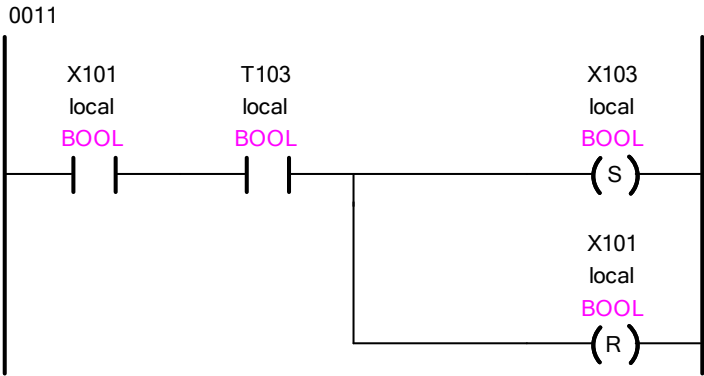
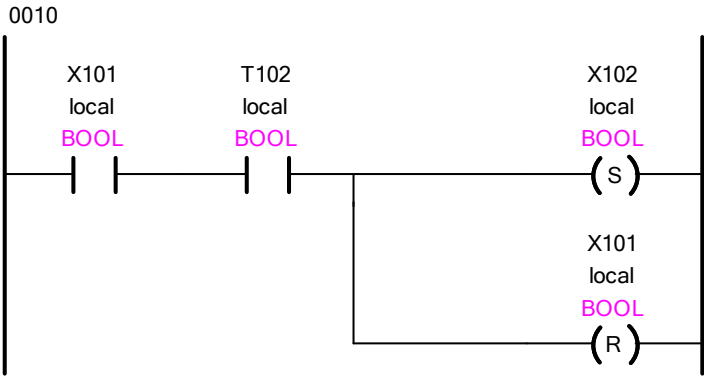
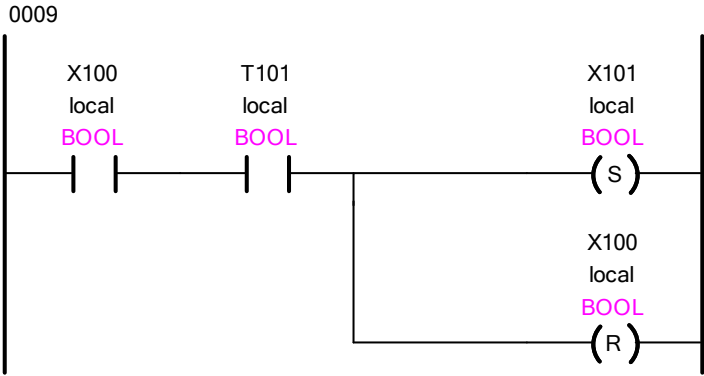
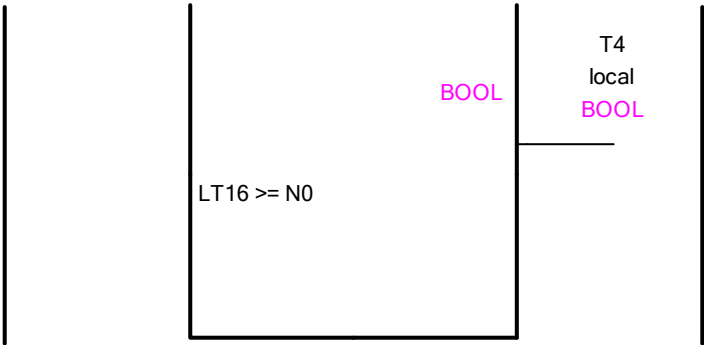
1.10 \\Logical\\EXPERIMENTAL\\Variables.var

Name	Reference	Type	Constant	Retention	Replicable	Value	Description [1]
 X100	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T104	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T105	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
 X1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 T1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N1	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TON_0	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TIM_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 BN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 bN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 Tn	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T5	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 v14	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 CONS2	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0.0	
 MTBasicsPID_1	<input type="checkbox"/>	MTBasicsPID	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	(PIDParameters:= (Gain:=2.334,Integration Time:=207.7))	
 LT16_REAL	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 SP	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 LT16_0	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 PID	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 SP_INT	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 ERROR	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 PID_INTEGER	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

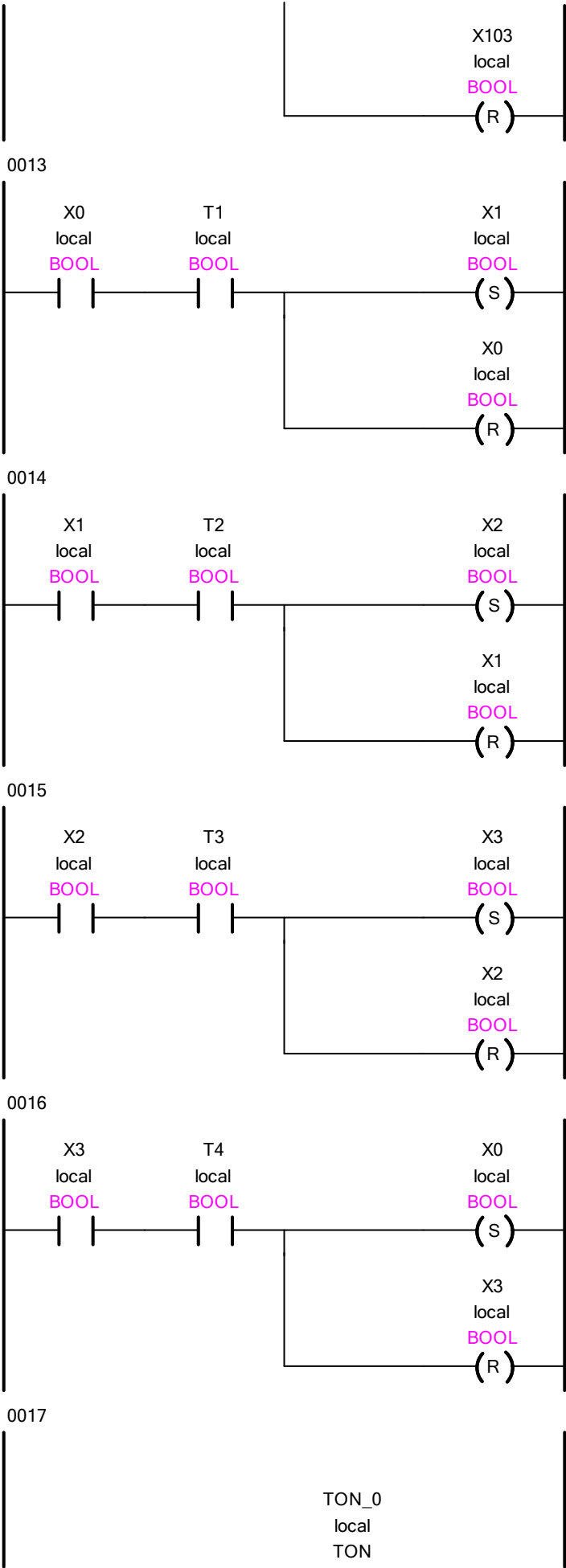
1.11 \Logical\ATURAR\_PE\Cyclic.ld

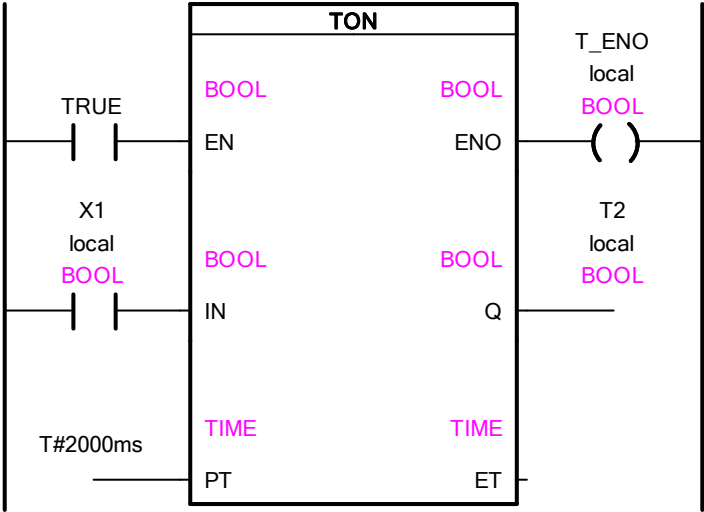
PROGRAM \_CYCLIC



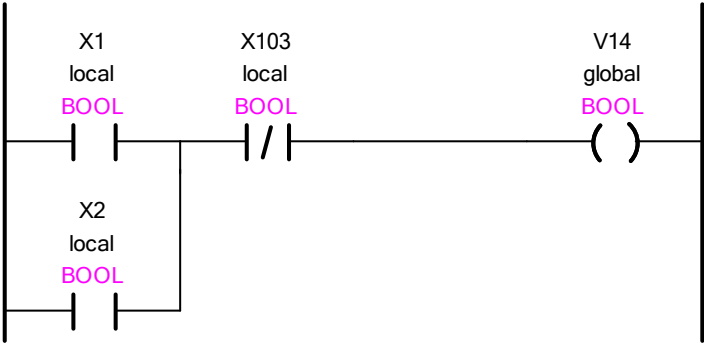




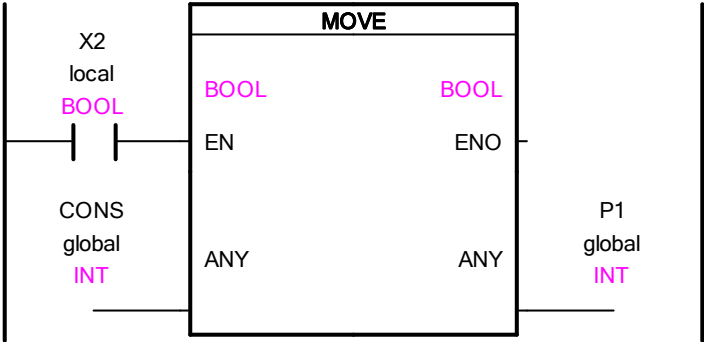




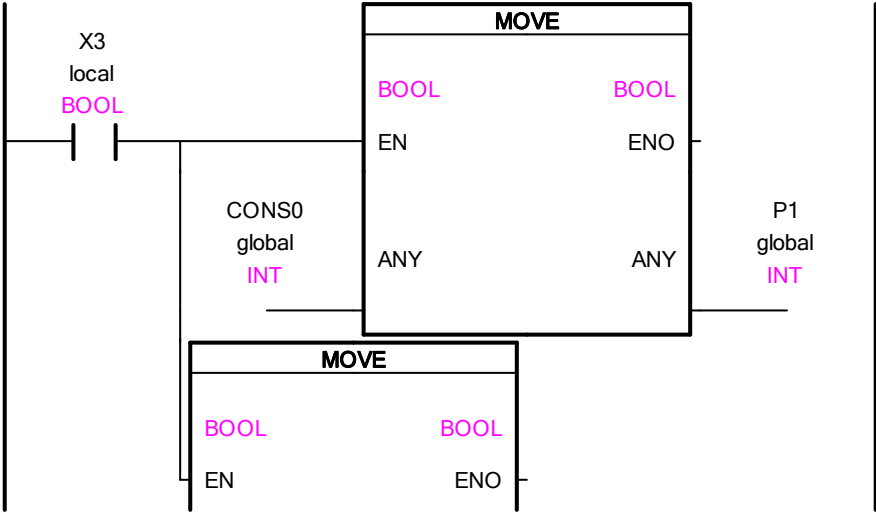
0018

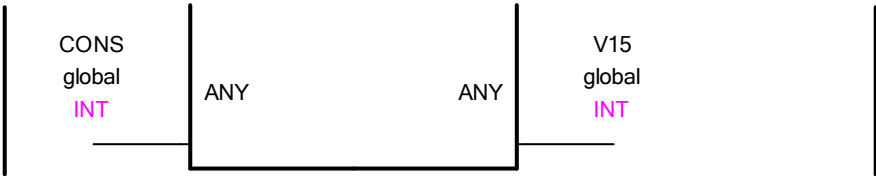


0019

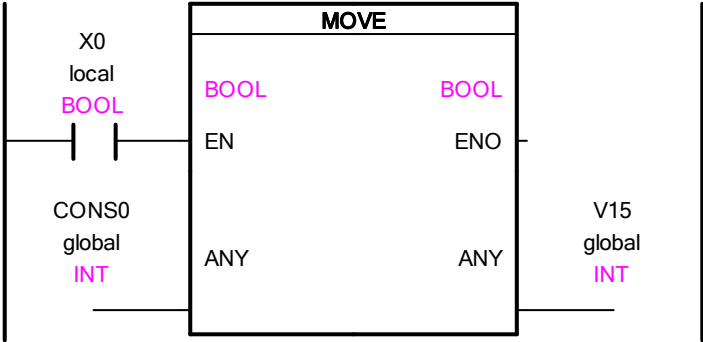


0020





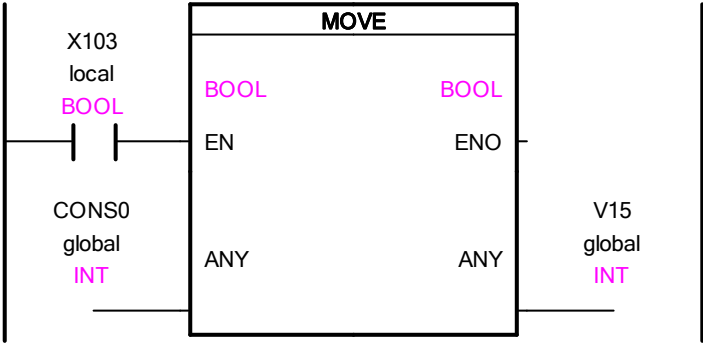
0021



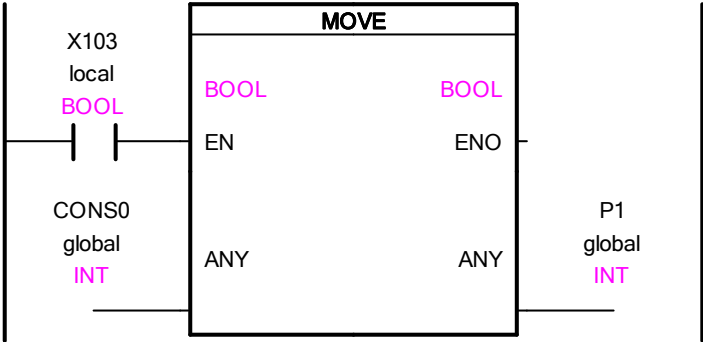
0022



0023

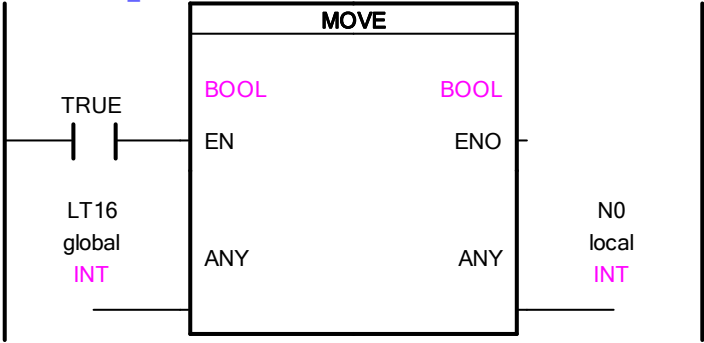


0024



1.12 \Logical\ATURAR\_PE\Init.Id



























0001  
PROGRAM\_INIT



0002

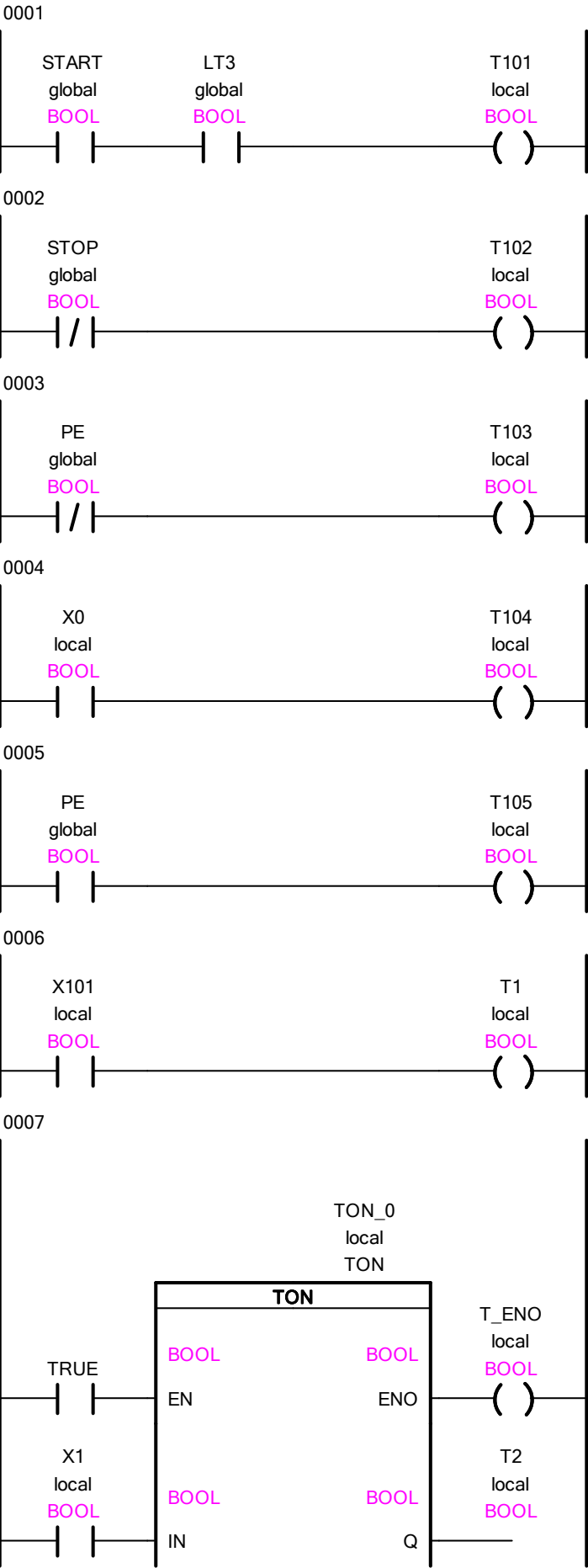


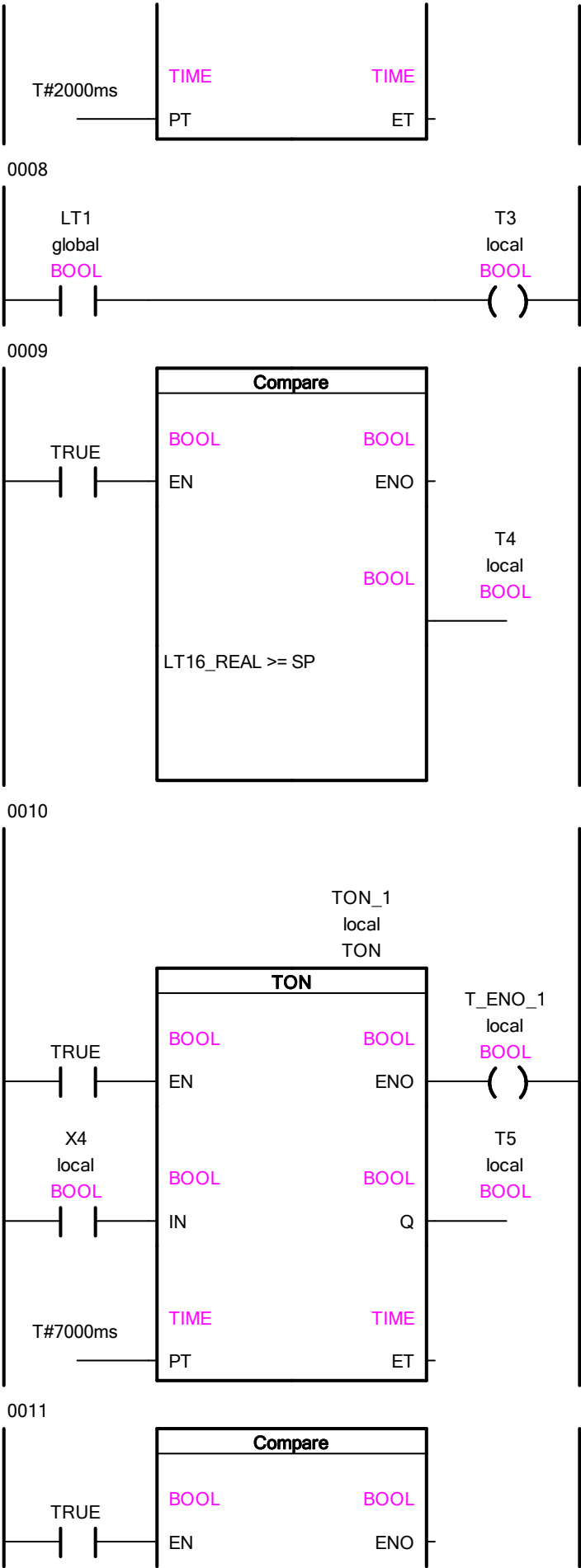
## 1.13 \\Logical\\ATURAR\_PE\\Variables.var

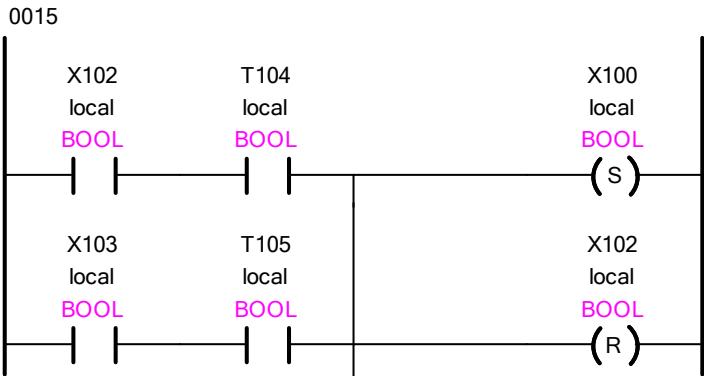
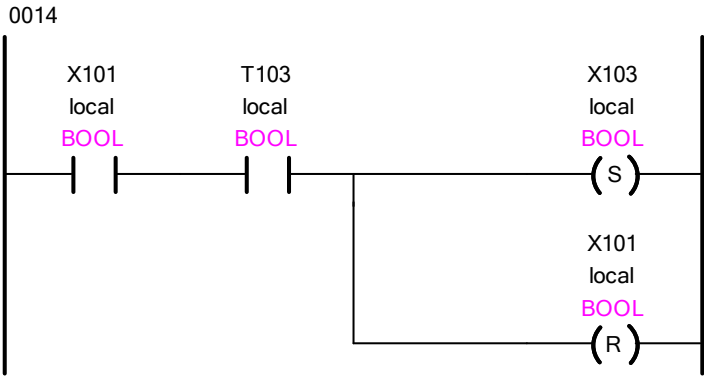
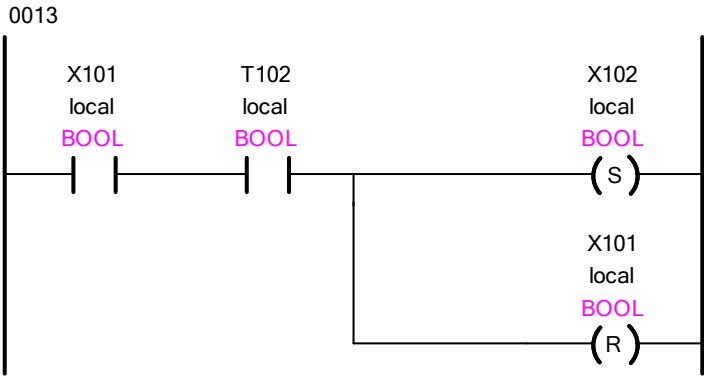
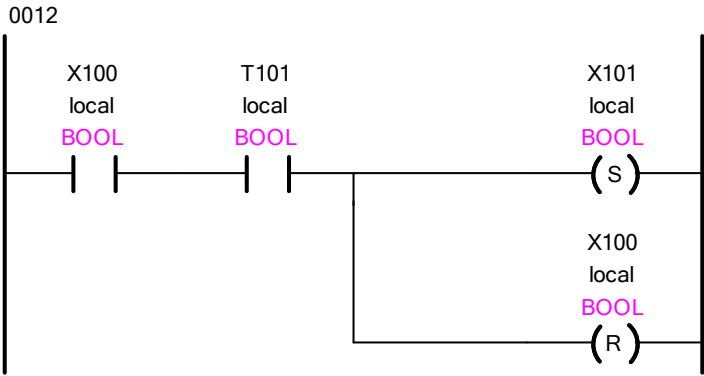
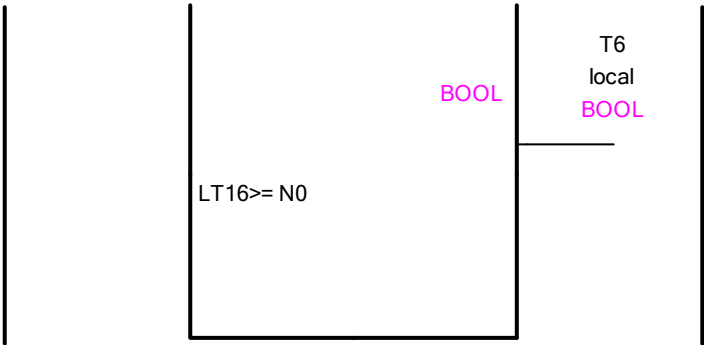
Name	Reference	Type	Constant	Retention	Replicable	Value	Description [1]
 X100	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T104	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T105	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
 X1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 T1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N1	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N0	<input type="checkbox"/>	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32500	
 TON_0	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TIM_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 BN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 bN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T5	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 LCPID_0	<input type="checkbox"/>	LCPID	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		

1.14 \Logical\PID\Cyclic.Id

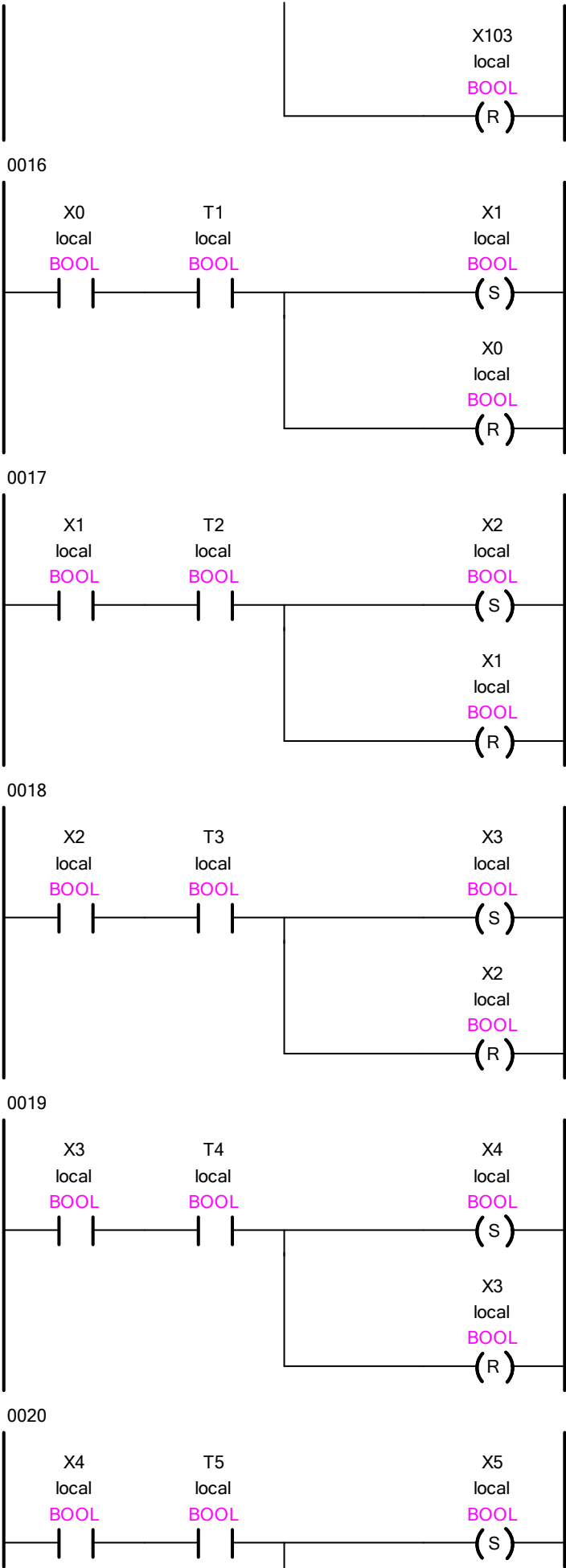
PROGRAM \_CYCLIC

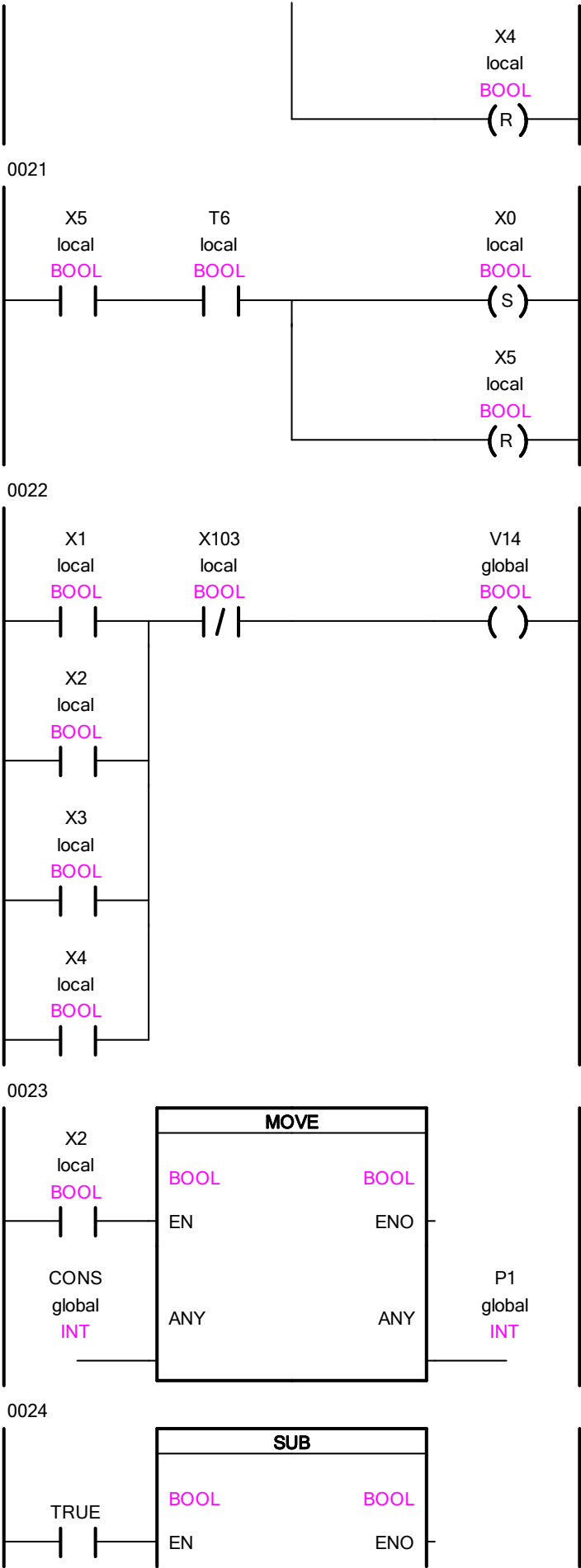


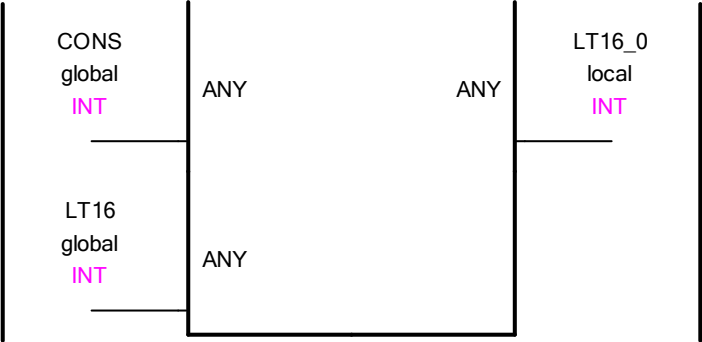




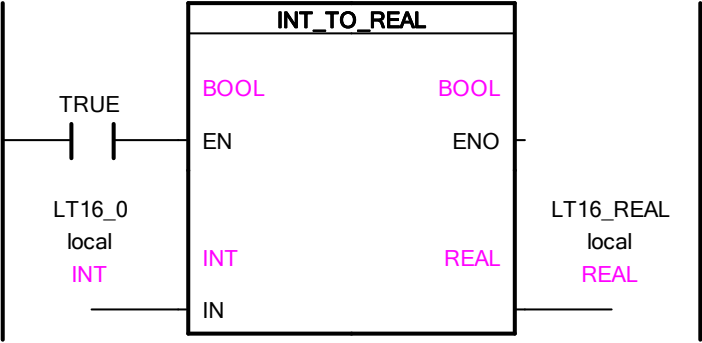




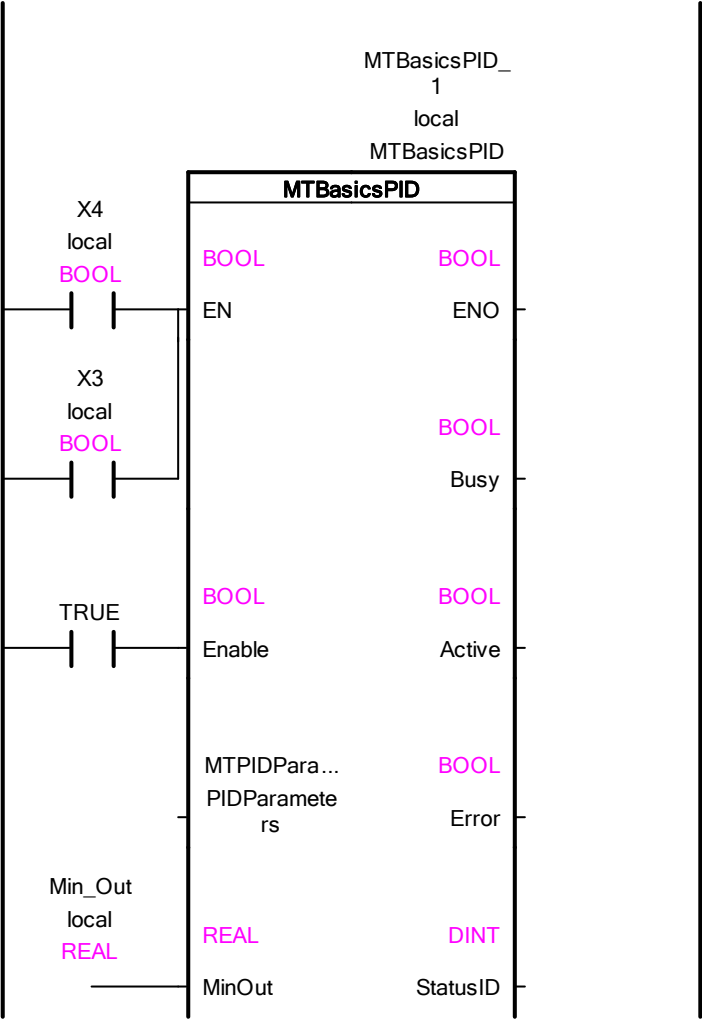


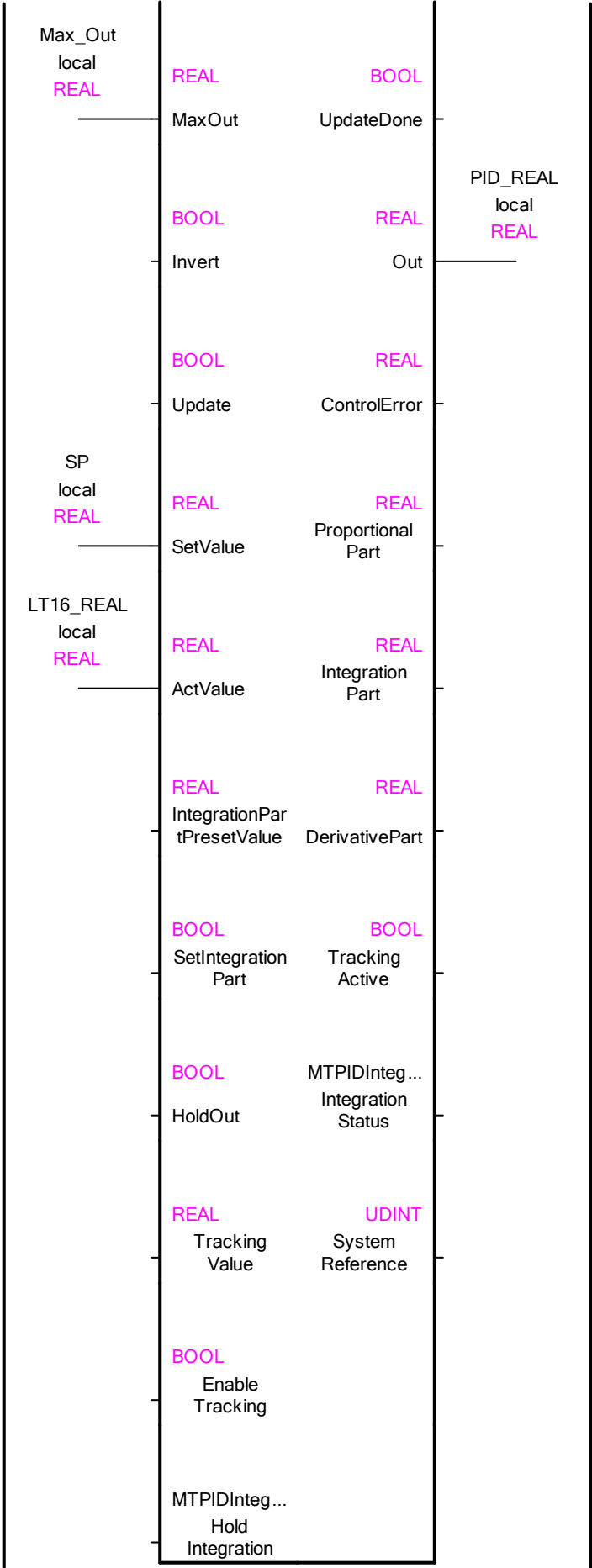


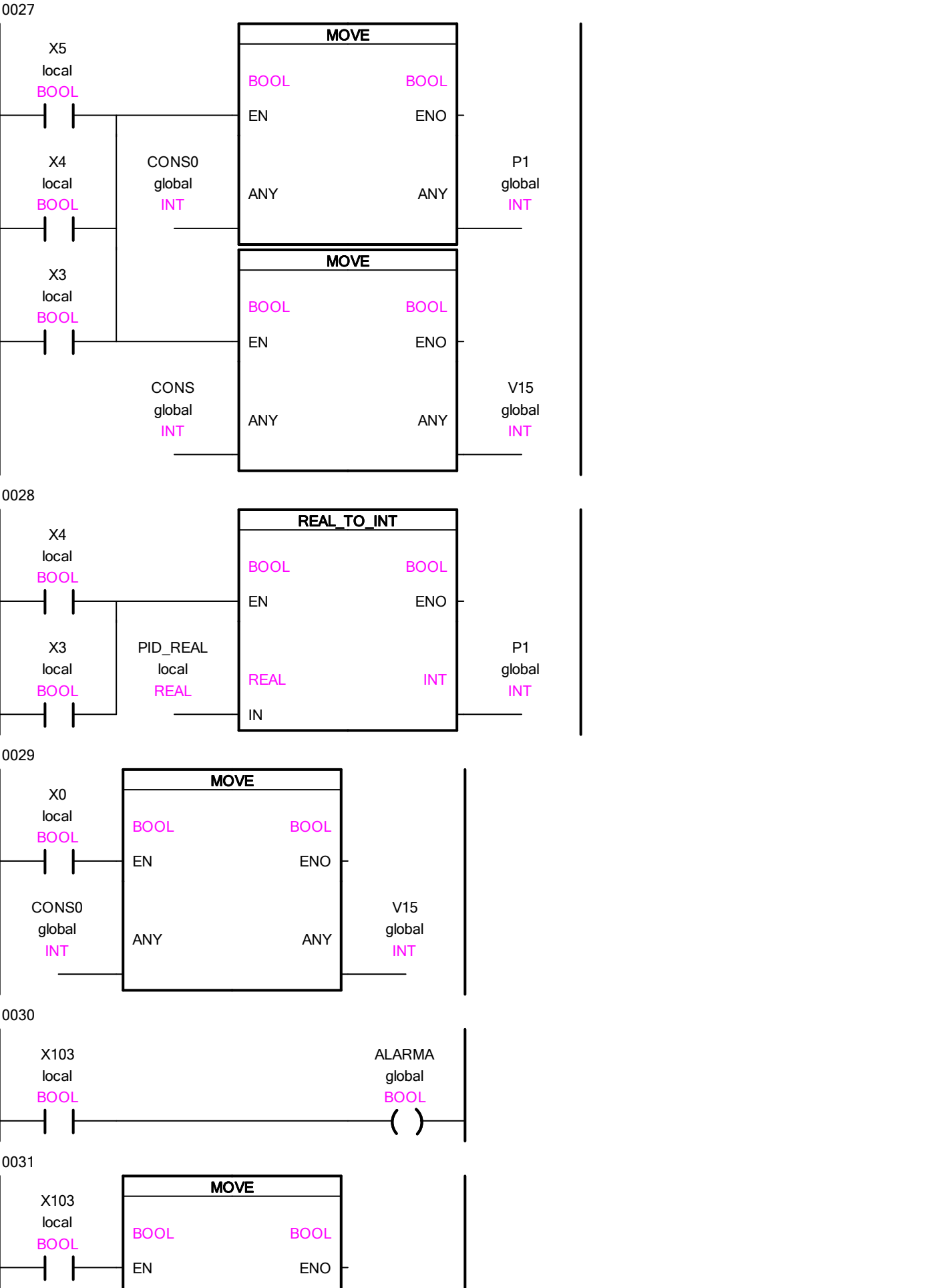
0025



0026

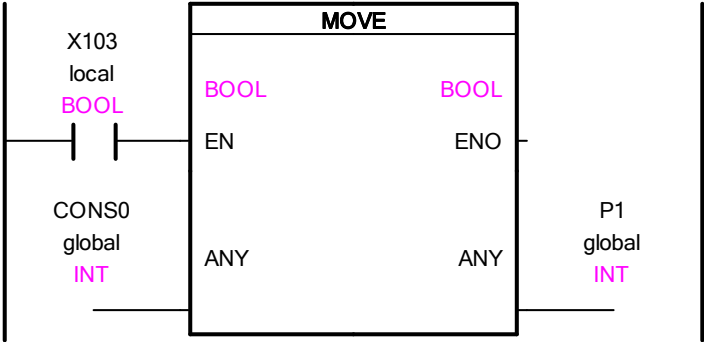




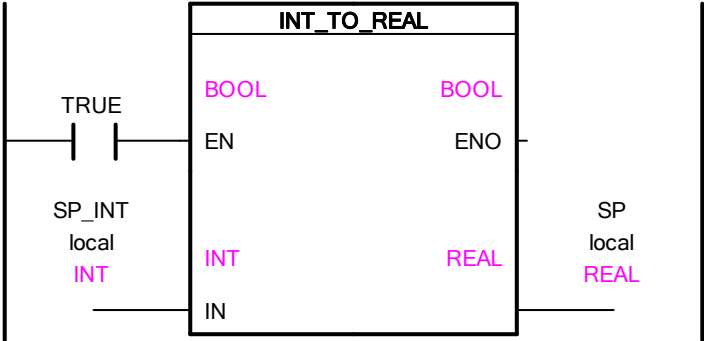




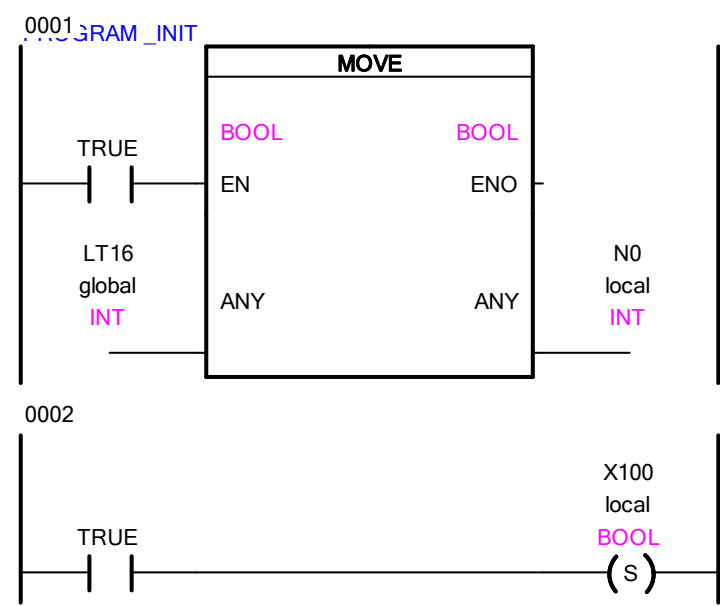
0032









































0033



1.15 \Logical\PID\Init.Id



1.16 \\Logical\\PID\\Variables.var

Name	Reference	Type	Constant	Retention	Replicable	Value	Description [1]
 LT16_REAL	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 PID_REAL	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 LT16_0	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X100	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T101	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T102	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T103	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T104	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T105	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	TRUE	
 X1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 X3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	
 T1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T2	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T3	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N1	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 N0	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32500	
 TON_0	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TIM_ENO	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 BN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 bN0	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T5	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 X4	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 Max_Out	<input type="checkbox"/>	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	32000	
 SP	<input type="checkbox"/>	REAL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 SP_INT	<input type="checkbox"/>	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 Min_Out	<input type="checkbox"/>	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0	
 X5	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 MTBasicsPID_1	<input type="checkbox"/>	MTBasicsPID	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T6	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 TON_1	<input type="checkbox"/>	TON	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
 T_ENO_1	<input type="checkbox"/>	BOOL	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		



1.17 \Logical\Visu\VCOject.vc

Pages  
Page 1: Init\_Page (Default Page)

START

ATURAR

PE

LECTURA SENSOR NIVELL ANALÒGIC LT 1.6

0

CONSIGNA

0



















LT1



















LT3




















Used datapoints for page: Init\_Page

There are no datapoints connected to this page.



















Virtual key settings for page: Init\_Page

Name	Type	Details	Layer	Description
 TP_ALPHA_0		used globally		Key for input '0'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(0)		
 TP_ALPHA_1		used globally		Key for input '1'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(1)		
 TP_ALPHA_2		used globally		Key for input '2'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(2)		
 TP_ALPHA_3		used globally		Key for input '3'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(3)		
 TP_ALPHA_4		used globally		Key for input '4'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(4)		
 TP_ALPHA_5		used globally		Key for input '5'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(5)		
 TP_ALPHA_6		used globally		Key for input '6'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(6)		
 TP_ALPHA_7		used globally		Key for input '7'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(7)		
 TP_ALPHA_8		used globally		Key for input '8'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(8)		
 TP_ALPHA_9		used globally		Key for input '9'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(9)		
 TP_ALPHA_A		used globally		Key for input 'A'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(A)		
 TP_ALPHA_Ampersand		used globally		Key for input '&'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(&)		
 TP_ALPHA_Apostrophy		used globally		Key for input '''
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(')		
 TP_ALPHA_Apostrophy1		used globally		Key for input ''''
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(")		
 TP_ALPHA_Asterisk		used globally		Key for input '*'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(*)		
 TP_ALPHA_At_sign		used globally		Key for input '@'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(@)		
 TP_ALPHA_B		used globally		Key for input 'B'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(B)		
 TP_ALPHA_Backquote		used globally		Key for input '`'
◆ Default	<no type>		Default	

Name ▲	Type	Details	Layer	Description
◆ Global actions	InputCharacter	Char(')		
 TP_ALPHA_BckSlsh		used globally		Key for input '\'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(\)		
 TP_ALPHA_Bigger		used globally		Key for input '>'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(>)		
 TP_ALPHA_C		used globally		Key for input 'C'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(C)		
 TP_ALPHA_Care		used globally		Key for input '^'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(^)		
 TP_ALPHA_Closebrace		used globally		Key for input '}'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(})		
 TP_ALPHA_Closebracket		used globally		Key for input ']'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char()])		
 TP_ALPHA_Closeparen		used globally		Key for input ')'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char())		
 TP_ALPHA_Colon		used globally		Key for input ':'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(:)		
 TP_ALPHA_Comma		used globally		Key for input ','
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(,)		
 TP_ALPHA_D		used globally		Key for input 'D'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(D)		
 TP_ALPHA_Dollar		used globally		Key for input '\$'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(\$)		
 TP_ALPHA_DOT		used globally		Key for input '.'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(.)		
 TP_ALPHA_E		used globally		Key for input 'E'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(E)		
 TP_ALPHA_Equal		used globally		Key for input '='
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(=)		
 TP_ALPHA_Exclamation		used globally		Key for input '!'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(!)		
 TP_ALPHA_F		used globally		Key for input 'F'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(F)		
 TP_ALPHA_G		used globally		Key for input 'G'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(G)		
 TP_ALPHA_H		used globally		Key for input 'H'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(H)		

Name ▲	Type	Details	Layer	Description
 TP_ALPHA_Hash		used globally		Key for input '#'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(#)		
 TP_ALPHA_I		used globally		Key for input 'I'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(I)		
 TP_ALPHA_J		used globally		Key for input 'J'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(J)		
 TP_ALPHA_K		used globally		Key for input 'K'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(K)		
 TP_ALPHA_L		used globally		Key for input 'L'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(L)		
 TP_ALPHA_M		used globally		Key for input 'M'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(M)		
 TP_ALPHA_Minus		used globally		Key for input '-'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(-)		
 TP_ALPHA_N		used globally		Key for input 'N'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(N)		
 TP_ALPHA_O		used globally		Key for input 'O'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(O)		
 TP_ALPHA_Openbrace		used globally		Key for input '{'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char({)		
 TP_ALPHA_Openbracket		used globally		Key for input '['
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char([)		
 TP_ALPHA_Openparen		used globally		Key for input '('
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char((		
 TP_ALPHA_P		used globally		Key for input 'P'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(P)		
 TP_ALPHA_Percent		used globally		Key for input '%'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(%)		
 TP_ALPHA_Plus		used globally		Key for input '+'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(+)		
 TP_ALPHA_Q		used globally		Key for input 'Q'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(Q)		
 TP_ALPHA_Quest		used globally		Key for input '?'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(?)		
 TP_ALPHA_Quotes		used globally		Key for input '"'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(")		
 TP_ALPHA_R		used globally		Key for input 'R'

Name ▲	Type	Details	Layer	Description
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(R)		
TP_ALPHA_S		used globally		Key for input 'S'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(S)		
TP_ALPHA_sa		used globally		Key for input 'a'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(a)		
TP_ALPHA_sb		used globally		Key for input 'b'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(b)		
TP_ALPHA_sc		used globally		Key for input 'c'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(c)		
TP_ALPHA_sd		used globally		Key for input 'd'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(d)		
TP_ALPHA_se		used globally		Key for input 'e'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(e)		
TP_ALPHA_Semicolon		used globally		Key for input ';'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(;		
TP_ALPHA_sf		used globally		Key for input 'f'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(f)		
TP_ALPHA_sg		used globally		Key for input 'g'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(g)		
TP_ALPHA_sh		used globally		Key for input 'h'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(h)		
TP_ALPHA_si		used globally		Key for input 'i'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(i)		
TP_ALPHA_sj		used globally		Key for input 'j'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(j)		
TP_ALPHA_sk		used globally		Key for input 'k'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(k)		
TP_ALPHA_sl		used globally		Key for input 'l'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(l)		
TP_ALPHA_Slash		used globally		Key for input '/'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(/)		
TP_ALPHA_sm		used globally		Key for input 'm'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(m)		
TP_ALPHA_Small		used globally		Key for input '<'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(<)		
TP_ALPHA_sn		used globally		Key for input 'n'
◆ Default	<no type>		Default	

Name ▲	Type	Details	Layer	Description
◆ Global actions	InputCharacter	Char(n)		
 TP_ALPHA_so		used globally		Key for input 'o'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(o)		
 TP_ALPHA_sp		used globally		Key for input 'p'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(p)		
 TP_ALPHA_Space		used globally		Key for input ' '
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char( )		
 TP_ALPHA_sq		used globally		Key for input 'q'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(q)		
 TP_ALPHA_sr		used globally		Key for input 'r'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(r)		
 TP_ALPHA_ss		used globally		Key for input 's'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(s)		
 TP_ALPHA_st		used globally		Key for input 't'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(t)		
 TP_ALPHA_su		used globally		Key for input 'u'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(u)		
 TP_ALPHA_sv		used globally		Key for input 'v'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(v)		
 TP_ALPHA_sw		used globally		Key for input 'w'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(w)		
 TP_ALPHA_sx		used globally		Key for input 'x'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(x)		
 TP_ALPHA_sy		used globally		Key for input 'y'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(y)		
 TP_ALPHA_sz		used globally		Key for input 'z'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(z)		
 TP_ALPHA_T		used globally		Key for input 'T'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(T)		
 TP_ALPHA_Tilde		used globally		Key for input '~'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(~)		
 TP_ALPHA_U		used globally		Key for input 'U'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(U)		
 TP_ALPHA_Underscore		used globally		Key for input '_'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(_)		
 TP_ALPHA_V		used globally		Key for input 'V'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(V)		

Name ▲	Type	Details	Layer	Description
 TP_ALPHA_VBar		used globally		Key for input 'I'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(I)		
 TP_ALPHA_W		used globally		Key for input 'W'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(W)		
 TP_ALPHA_X		used globally		Key for input 'X'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(X)		
 TP_ALPHA_Y		used globally		Key for input 'Y'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(Y)		
 TP_ALPHA_Z		used globally		Key for input 'Z'
◆ Default	<no type>		Default	
◆ Global actions	InputCharacter	Char(Z)		
 TP_BACKSPACE		globally exclusive		Key for backspace
◆ Default	<no type>		Default	
◆ Global actions	InputControl	Action(Backspace)		
 TP_CANCEL		globally exclusive		Key for cancel input
◆ Default	<no type>		Default	
◆ Global actions	InputControl	Action(Cancel Input)		
 TP_Confirm		globally exclusive		Key for confirm input
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Numeric(Confirm Input),Listbox(Confirm Input),Alarm(Enter),String(Confirm Input),Password(Confirm Input),Edit(Confirm Input),Dropdown(Confirm Input))		
 TP_Copy		used globally		Key for copy
◆ Default	<no type>		Default	
◆ Global actions	Clipboard	Mode(Copy)		
 TP_Cur_Down		globally exclusive		Key for move cursor down
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Down,Item)		
 TP_Cur_Left		globally exclusive		Key for move cursor left
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Left,Item)		
 TP_Cur_Right		globally exclusive		Key for move cursor right
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Right,Item)		
 TP_Cur_Up		globally exclusive		Key for move cursor up
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Up,Item)		
 TP_Cut		used globally		Key for cut
◆ Default	<no type>		Default	
◆ Global actions	Clipboard	Mode(Cut)		
 TP_End		globally exclusive		Key for move cursor to end of line
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Right,End)		
 TP_ENTER		globally exclusive		Key for input enter
◆ Default	<no type>		Default	
◆ Global actions	InputControl	Action(Enter)		

Name ▲	Type	Details	Layer	Description
 TP_Home		globally exclusive		Key for move cursor to start of line
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Left,End)		
 TP_Insert		globally exclusive		Key for toggle insert mode
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Edit(Toggle Insert Mode))		
 TP_NaviPadChancel		globally exclusive		Focus depending chancel
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Numeric(Cancel Input),Listbox(Cancel Input),Alarm(Cancel Input),String(Cancel Input),Password(Cancel Input))		
 TP_NaviPadDown		globally exclusive		Focus depending line down
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Numeric(Step Down),Listbox(Line Down),Alarm(Line Down))		
 TP_NaviPadEnter		globally exclusive		Focus depending enter
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Numeric(Enter),Listbox(Enter),Alarm(Acknowledge),String(Enter),Password(Enter))		
 TP_NaviPadUP		globally exclusive		Focus depending line up
◆ Default	<no type>		Default	
◆ Global actions	FocusDependent	Action(Numeric(Step Up),Listbox(Line Up),Alarm(Line Up))		
 TP_Page_Down		globally exclusive		Key for move one page down
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Down,Page)		
 TP_Page_Up		globally exclusive		Key for move one page up
◆ Default	<no type>		Default	
◆ Global actions	MoveCursor	Direction(Up,Page)		
 TP_Paste		used globally		Key for insert clipboard
◆ Default	<no type>		Default	
◆ Global actions	Clipboard	Mode(Paste)		
 TP_SelectionMode		used globally		Key for switch mode between insert and select
◆ Default	<no type>		Default	
◆ Global actions	Clipboard	Mode(Toggle Selection Mode)		
 TP_SET_LAYER0		globally exclusive		Switch to Layer 0
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,0)		
 TP_SET_LAYER2		globally exclusive		Switch to Layer 2
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,2)		
 TP_SET_LAYER4		globally exclusive		Switch to Layer 4
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,4)		



Name ▲	Type	Details	Layer	Description
 TP_Shift0		globally exclusive		Decrement Key Layer for one character
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Decrement Single)		
 TP_Shift1		globally exclusive		Increment Key Layer for one character
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Increment Single)		
 TP_Shift2		globally exclusive		Switch to Layer 0
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,0)		
 TP_Shift3		globally exclusive		Switch to Layer 2
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,2)		
 TP_Shift4		globally exclusive		Switch to Layer 4
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,4)		
 TP_ShiftCapsLock0		globally exclusive		Switch to Layer 1 - for alpha pad
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,1)		
 TP_ShiftCapsLock1		globally exclusive		Switch to Layer 0 - for alpha pad
◆ Default	<no type>		Default	
◆ Global actions	ChangeKeyLevel	Mode(Set,0)		
 TP_SIGN_CHANGE		globally exclusive		Change of sign at input control
◆ Default	<no type>		Default	
◆ Global actions	InputControl	Action(Negative Sign)		
 TP_TAB		globally exclusive		Key for insert tabulator
◆ Default	<no type>		Default	
◆ Global actions	InputControl	Action(Tabulator)		

There are no commonlayers defined in the panel.